Office de la propriété intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An Agency of Industry Canada

Bureau canadien des brevets

Certification

Canadian Patent Office

Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,337,642**, on February 21, 2001, by **PMC-SIERRA, INC.**, assignee of Carl D. McCrosky, Bernard Guay, Douglas Konkin, Steven F. Lang and Winston K. Mok, for "A Bus Interface for Transfer of Multiple SONET/SDH Rates over a Serial Backplane".

Agent certificateur/Certifying Officer

January 19, 2006
Date

Canada

(CIPO 68)
31-03-04

OPIC CIPO

A BUS INTERFACE FOR TRANSFER OF MULTIPLE SONET/SDH RATES OVER A SERIAL BACKPLANE

BACKGROUND

SONET/SDH is being employed in a wide range of contemporary telecommunications architectures, both in its traditional role as a protected Layer 1 solution in SONET rings and drops, and as a simpler framing layer protocol in more recent Intelligent Optical Network solutions (in which traditional SONET protection is replaced by other, higher layer solutions, but SONET framing is retained). In both of these uses, the SONET bandwidth necessary in individual network products is growing rapidly, and forcing lower power, denser solutions. One of the key techniques appropriate to meet these demands is the use of serial bus technologies and protocols to replace the older parallel, explicitly clocked buses, such as the Combus parallel bus defined for SONET at STS-3. The use of fast serial protocols: reduces pin counts and backplane trace counts, simplifies system timing by having Rx modules recover clock and data, reduces overall power consumption.

A prior PMC patent application describes a serial SONET/SDH solution for STS-12, where 8b/10b coding is used to ensure a adequate transition density for clock and data recovery in the Rx, and to mark SONET framing and pointers. The use of 8b/10b coding simplifies the Rx design, but increases the data rate from (the inherent) 622.08Mb/s to 777.6Mb/s (which is (10/8)*622.08).

The present work includes several significant advances on the earlier 777.6Mb/s solution:

1) the ability to carry STS-12, STS-24, STS-48, and STS-51,
2) the ability to use either or both: scrambled NRZ and 8b/10b coding,
3) the ability to use PRBS generators and monitors to test either entire payloads, or STS-Nc's.
4) the ability to force framing errors in order to test links.

The goal of the invention is to provide a serial method to connect various SONET/SDH termination and processing devices. The protocol should support the following SONET/SDH flows: the earlier 777.6 / 8b/10 solution for STS-12, and scrambled SONET/SDN at STS-12 (622.08Mb/s), STS-24 (1244.16Mb/s), STS-48 (2488.32Mb/s), and STS-51 (2643.84Mb/s. Each such SONET/SDH flow should be transmitted in duplex over a single pair of differential traces in each direction; each such duplex flow should be able to be configured for any of the above line rates and codings. The differential signaling techniques used should be based on LVDS-like methods. The power consumed by the implementation should be minimal. The protocol should include several options for manipulation of SONET/SDH Section, Line, and Path overhead octets; it should also support line testing via PRBS techniques.

Further goals relate to providing flexibility in receiver alignment and overall fabric timing: Each receiver must find both byte and SONET frame alignment. A device with multiple receivers must be capable of finding a mutual alignment of the frames on all receivers. Also, a device with multiple receivers must be capable of dividing it receivers into two

groups, which achieve separate mutual alignments. Finally, the receiver must support a sufficient depth of fifo to allow incoming signals in the same mutual alignment group to have up to two SONET frames differential delay on their paths, to allow some signals to entirely skip space switching stages, while other mutually aligned signals pass through the space stage.

DESCRIPTION

PROTOCOL DESCRIPTION

The essential elements of the serial bus protocol covered by this patent are:

1) SONET/SDH frames are carried in a one-to-one mapping with line codes,
2) SONET/SDH (standard and pseudo-standard) rates of STS-12, STS-24, STS-48, and STS-51 are suported.
3) A1 and A2 overheads my be: a) consistently overwritten with the correct alignment values, b) mis-written once (for error checking), or c) mis-written consistently (for error checking).
4) J0 can optionally be overwritten.
5) PRBS values can fill the entire (non-overhead) contents of a serial link, or of a STS-Nc.
6) The PRBS stream can be consistently negated.
7) B1 can be overwritten with standard BIP values, or a single BIP error can be signaled.
8) The byte stream can be 8b/10b encoded.
9) The resultant byte stream is serialized for transmission over a differential pair.

FUNCTIONAL DESCRIPTION

The function description of this invention is presented in two major sections, one covering the transmitter (Tx) module, the other covering the receiver module (Rx).

1.1 Transmitter Definition

1.1.1 FEATURES

- Receives SONET STS-12/48/51 frames on a nine bit interface (eight bits of data and one bit of control).
- Optionally Serial TeleCombus 8B/10B encodes the received SONET STS-12/48/51 frame.
- Optionally Serial Non-Return to Zero (S-NRZ) scrambles the received SONET STS-12/48/51 frame.
- Optionally first scrambles and then Serial TeleCombus 8B/10B encodes the recieved SONET STS-12/48/51 frame.

- Optionally overwrites the A1 and A2 byte positions with the SONET specified A1 and A2 values respectively.
- Optionally overwrites the A1 and A2 byte positions with an erred version of the A1 and A2 byte values. This option may be done for a single SONET STS-N frame or continuously to all SONET STS-N frames.
- Software configurable J0 character insertion into the 8B/10B encoded output data stream's J0 position.
- Recognizes a special line code violation (LCV) character on the ingress data stream and maps this to a LCV on the egress data stream.
- Optionally inserts a single, or a multiple, configurable test pattern on the egress data stream.
- Optionally inserts a raw pseudo-random bit sequence (PRBS), using the $x^{23}+x^{18}+1$ generator polynomial, on the egress data stream.
- Optionally inserts a PRBS, using the $x^{23}+x^{18}+1$ generator polynomial, within a SONET STS-Nc synchronous payload envelope (SPE), with the J1 byte in a fixed position, on the egress data stream. The H1 and H2 bytes of this STS-Nc frame are correctly encode to indicate a concatenated frame format.
- Optionally inverts the PRBS on the egress data stream.
- Optionally inserts Bit Interleaved Parity (BIP-8) calculated bytes into the B1 byte position of the first STS-1 of an SONET STS-N frame.
- Optionally inserts a single BIP-8 error into the B1 byte position.
- Optionally inverts the output data stream (OD[9:0]).
- Provides wrapper functionality for PISO-1250 and TXLV-1250 ABCs.

1.1.2 OPERATION

1.1.2.1 A1/A2 Insertion Procedure

To enable insertion of SONET standard A1/A2 characters into the egress data stream write a logic one to the AINS bit (bit zero of the TSEC Control and Status register).

To disable insertion of the SONET standard A1/A2 characters into the egress data stream write a logic zero to the AINS bit of the TSEC Control and Status

register. The default value for the AINS bit is logic zero (disabling insertion of SONET standard A1/A2 characters).

To enable insertion of A1/A2 character errors into the egress data stream write a logic one to the AINS bit (bit zero of the TSEC Control and Status register) and write a logic one to the A1A2ERR_CONT bit (bit 14 of the TSEC Test Pattern and Error Insertion Control register).

To enable insertion of A1/A2 characters errors into only the next SONET STS-N frame write a logic one to the AINS bit (bit zero of the TSEC Control and Status register) and write a logic one to the A1A2ERR_SINGLE bit (bit 13 of the TSEC Test Pattern and Error Insertion Control register). The A1A2ERR_SINGLE bit self clears once the insertion has taken place, enabling another single framing error to be inserted.

To enable insertion of A1/A2 characters errors into only the next SONET STS-N frame relative to the incoming frame pulse (IJ0) write a logic one to the AINS bit (bit zero of the TSEC Control and Status register) and write a logic one to the A1A2ERR_SINGLE bit (bit 13 of the TSEC Test Pattern and Error Insertion Control register). The A1A2ERR_SINGLE bit self clears once the insertion has taken place, enabling another single framing error to be inserted.

1.1.2.2 J0 Insertion Procedure

To enable insertion of the SONET standard J0 character into the egress data stream write a logic one to the J0INS bit (bit one) of normal mode register 00h (TSEC Control and Status register). To disable insertion of the SONET standard J0 character into the egress data stream write a logic zero to the J0INS bit (bit three of the TSEC Control and Status register). The default value for the J0INS bit is logic zero (disabling insertion of the SONET standard J0 character). J0 insertion is only valid when the TSEC is in Encoder and Encoder-Scrambler mode.

1.1.3 FUNCTIONAL TIMING

The following figure illustrates the input of a special character (J0 byte) to the TSEC while operating in the Encoder-Scrambler mode. Note that the J0 byte position is indicated by the IJ0 input signal.

Figure 1 – J0 Indication Input Signals In Encoder-Scrambler Mode

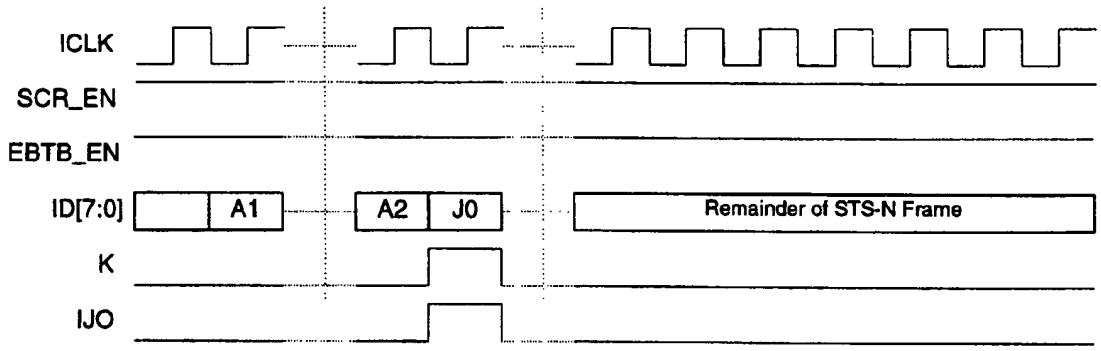


Figure 2 shows the input of a special character (J0 byte) to the TSEC while operating in the Encoder mode. Note that the internal character representing J0 is 00 hex with both K and IJ0 a logic one value. The output of the J0 byte while in Encoder mode is the 8B/10B Serial TeleCombus format of K28.5+/- dependent on the current running disparity.

Figure 2 – Special Character Input In Encoder Mode

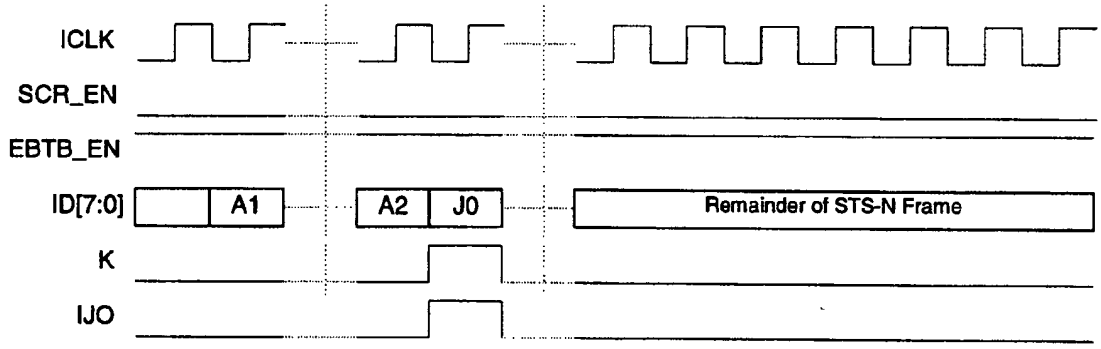


Figure 3 shows SONET aware PRBS Generator output for the STS-Nc payload types in Scrambler mode. The figure shows the IJ0 signal pulse high to identify the J0 byte. The J1 byte position marks the start of a new SPE. The output data shown (OD[8:1]) occurs after a fixed number of clock cycles and not instantaneously as implied by the timing diagram. The output bytes labeled FSx are the fixed stuff bytes of a SONET concatenated frame.

Figure 3: Scrambler Mode SONET Aware PRBS Output

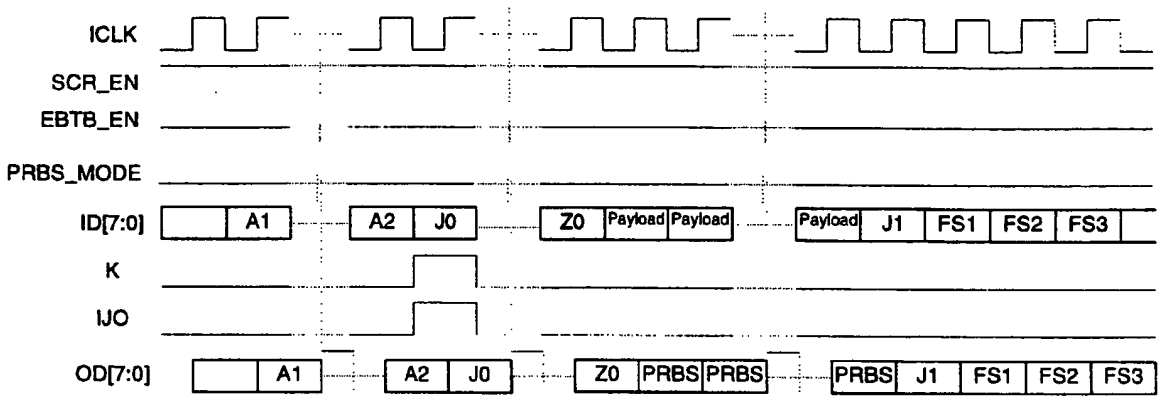
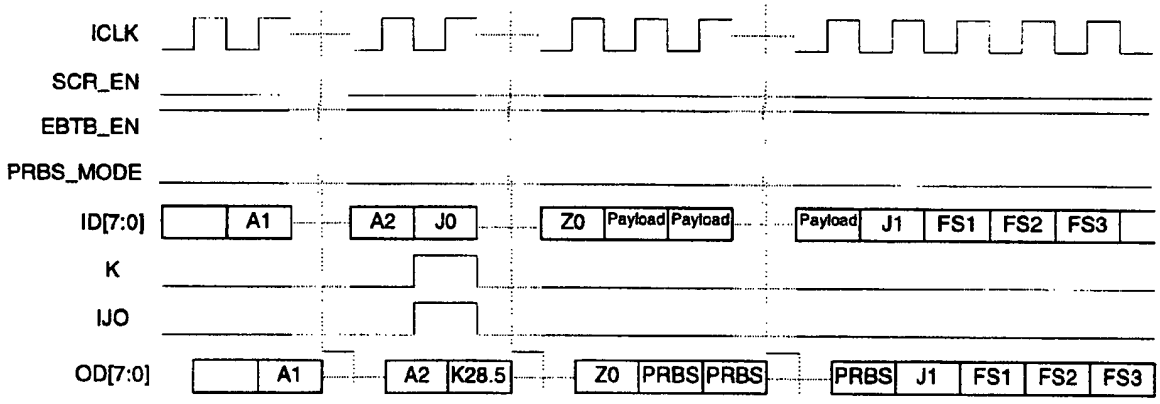


Figure 4 shows SONET aware PRBS Generator output for the STS-Nc payload types in Scrambler mode. The figure shows the IJ0 signal pulse high to identify the J0 byte. The J1 byte position marks the start of a new SPE. The output data shown (OD[8:1]) occurs after a fixed number of clock cycles and not instantaneously as implied by the timing diagram. The output bytes labeled FSx are the fixed stuff bytes of a SONET concatenated frame.

Figure 4: Encoder Mode SONET Aware PRBS Output



1.1.4 IMPLEMENTATION DESCRIPTION

1.1.4.1 Design Capture

The PM5146 TSB design is captured using VHDL and contains seven major blocks: the Character Inserter, the SONET Frame Counter, the PRBS Generator,

the Scrambler, the 8B/10B Encoder, the Output Select and the ECBI. The relationship between these blocks is shown in the TSEC.

1.1.4.1.1 SONET Frame Counter (frame_counter)

The Frame Counter block consists of three flywheel counters. A 4-bit counter tracks the current row of an STS-1 frame, a 7-bit counter tracks the current column of an STS-1 frame and a 6-bit counter counts the current STS-1 frame within an STS-N frame. The value of N for a given STS-N frame is specified by the STSMODE[1:0] input and can be either 12, 48, or 51. The Frame Counter outputs the data and k-character indication signal, as well as indications of the SPE (excluding path overhead and fixed stuff bytes), and the A1, A2, H1, H2, J0Z0, J0 and J1 byte positions, to the egress Character Inserter block.

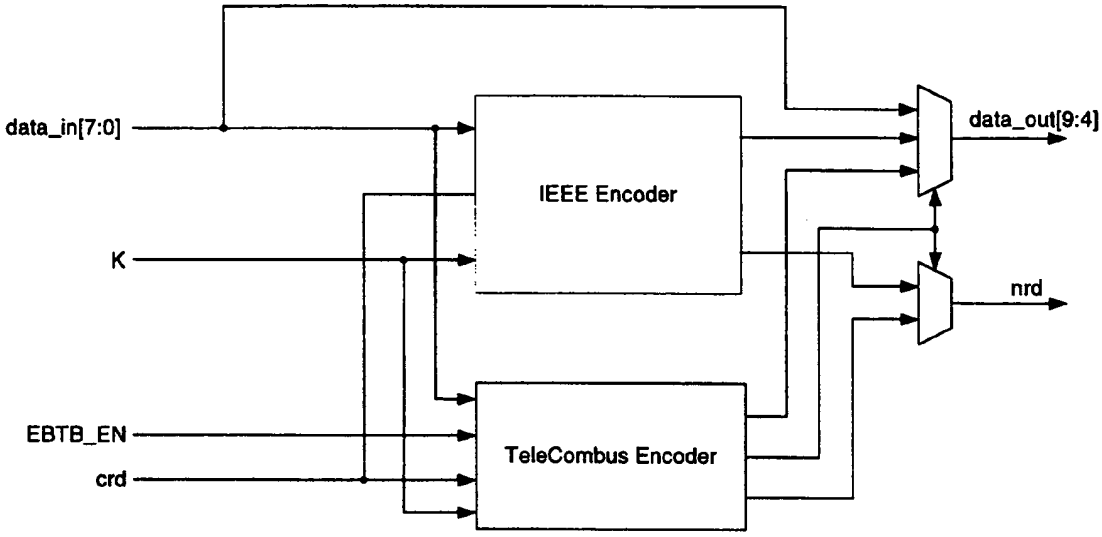
1.1.4.1.2 Character Inserter (char_inserter)

The Character Inserter block inserts the special A1, A2 and J0 characters of the SONET specification into the ingress data stream for the Scrambler block. The Character Inserter also inserts the BIP-8 value from the BIP8 block into the B1 byte position when B1 insertion is enabled. The Character Inserters' functionality is accomplished through an indication from the SONET Frame Counter of the J0, A1, A2, and B1 byte positions. The Character Inserter then inserts the appropriate byte depending on the functionality enabled. If the functionality is not enabled then the data passes through the Character Inserter block unaltered.

1.1.4.1.3 8B/10B Encoder Block (encoder_8b10b)

The 8B/10B Encoder block, shown in Figure 5, contains two main encoders.

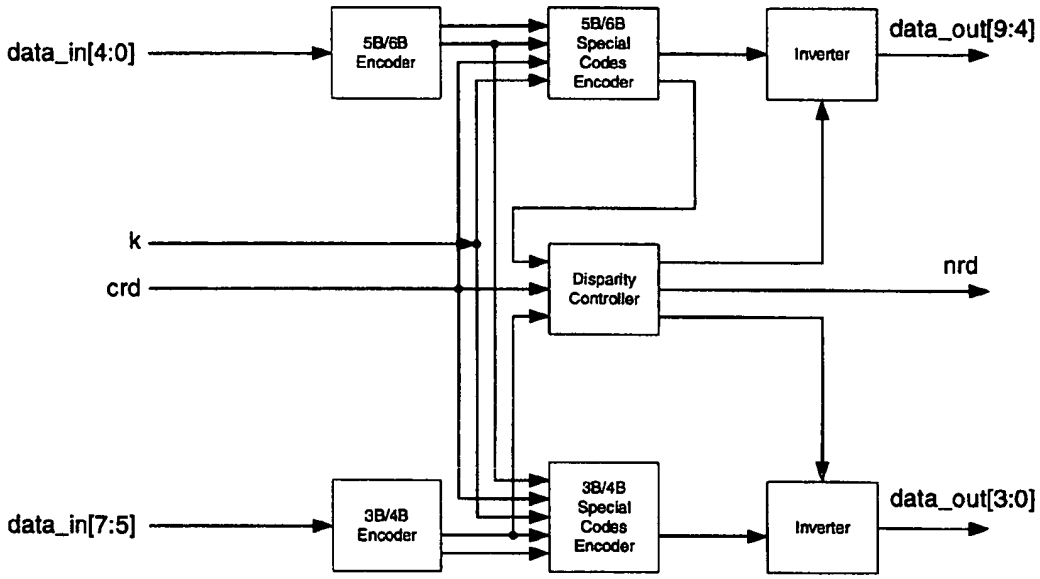
Figure 5 - 8B/10B Encoder



The first main block is the TeleCombus Encoder block. This block is a mapping of the TeleCombus signals, as well as an indication that the ingress byte is a TeleCombus control byte, a TeleCombus data byte, or a Scrambler mode byte, enabling the correct data to be placed on the egress data stream.

The second main block is an IEEE 8B/10B format encoder. This contains a 5B/6B Encoder, a 5B/10B Special Code Encoder, a 3B/4B Encoder, a 3B/4B Special Code Encoder, a Disparity Controller and Inverters as shown in Figure 6.

Figure 6 – IEEE 8B/10B Encoder



5B/6B Encoder: It first assumes the current running disparity is negative and encodes 5-bit data to 6-bit codes.

5B/6B Special Code Encoder: It deals with K control characters and some special cases for the 5-bit part.

3B/4B Encoder: It first assumes the current running disparity is negative and encodes 3-bit data to 4-bit codes.

3B/4B Special Code Encoder: It deals with K control characters and some special cases for the 3-bit part.

Disparity Controller: It controls whether the 6-bit or 4-bit encoded data needs to be inverted according to the actual current running disparity and the disparity of the encoded sub-blocks. Inversion of balanced sub-block codes is not allowed, no matter what the current disparity value is. If the current running disparity of the sub-block is positive, this sub-block code will be inverted. Table 1 shows the truth table of the Disparity Controller.

Table 1 - Disparity Controller Truth Table

CRD	FLIP_6B	RD at the End of 6-bit Sub-block	FLIP_4B	RD at the End of 4-bit Sub-block	INVEN_6B	INVEN_4B
0	0	0	0	0	0	0

0	0	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	0	0	1
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	1	1	0

CRD: current running disparity.

FLIP_XX: If the disparity of the sub-block is zero, FLIP_XX(XX can be 6B or 4B) flag will be set to zero. Otherwise, FLIP_XX is 1. Because the current disparity is assumed to be negative at first, the disparity of the encoded sub-block is definitely positive.

INVEN_XX: Invert enable signal for the 6-bit or 4-bit sub-block.

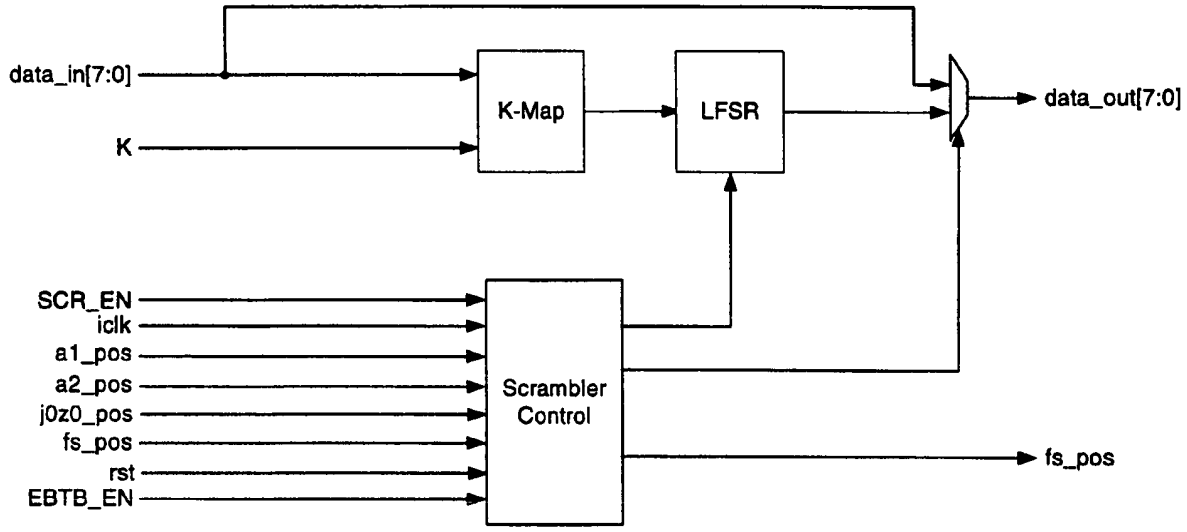
Inverter: It is a controlled inverter. When the control signal is high, the input code will be inverted.

The multiplexors on the data_in[7:5] and data_in[4:0] inputs enables the TSEC to place the 8B/10B Encoder block in a static, low-power, mode when this block is not utilized (SCRAMBLEMODE is a logic one).

1.1.4.1.4 Scrambler Block (snrz_scrambler)

The Scrambler block implements a standard S-NRZ Scrambler function. This block also has the functionality to convert the Serial TeleCombus 8B/10B Multiplex Section Termination Level format into scrambled output. The block diagram for the Scrambler is shown in Figure 7.

Figure 7 – Scrambler Block Diagram



The K-Map block converts the input data, along with the K input.. This is the block that converts the Serial TeleCombus MST 8B/10B format into the pre-scrambled output. Any data that is input to the Scrambler block that is not marked with a logic high on the K input is pre-scrambled data and requires no modifications before being scrambled.

The Scrambler Control block enables and resets the Scrambler block as described for ANSI standard scrambling of a SONET frame. The Scrambler block is reset to 7F hex on the first byte following the J0/Z0 column. The A1, A2, and J0/Z0 columns are not scrambled. Scrambler Control also selects, based on the EBTB_EN and SCR_EN inputs, the appropriate output from the Scrambler block.

The LFSR block performs the ANSI standard S-NRZ scrambling function. A functional diagram for this block is shown below. This figure describes the generator polynomial, $1 + x^6 + x^7$, implemented by the scrambler and is used for a serial input stream. To account for the parallel input data a logical exclusive-or of the input data bits 7 through 0 with the X_7 through X_0 bits is performed. Executing a parallel scrambling function requires an eight-bit increment in the scrambling sequence on each clock cycle. The logic required for this is shown below where an exclusive-or operation is performed on the bits in any inner cell of the table.

The output of the Scrambler block is an eight-bit wide parallel data stream.

1.1.4.1.5 BIP8 Block

The BIP8 block computes the BIP-8 byte value when BIP-8 insertion is enabled (bit five of the TSEC Status and Control register is logic high). The BIP-8 byte value is calculated per SONET STS-N frame. The first bit of the BIP-8 byte provides even parity over the first bit of all 8-bit sequences in the frame. The second bit provides even parity over the second bits of all 8-bit sequences in the frame and so on. Even parity is defined as having an even number of ones in each of all 8-bit sequences including the BIP-8 byte. The BIP-8 byte is calculated for the previous STS-N frame after scrambling and inserted in the B1 byte of the first STS-1 of the current STS-N frame before scrambling when in either Encoder-Scrambler or Scrambler mode. When in Encoder mode the BIP-8 is calculated on the pre-8B/10B encoded data of the previous SONET STS-N frame. The result is then inserted in the pre-8B/10B encoded B1 byte position of the first STS-1 of the current STS-N frame. In Encoder mode, special characters (indicated with the K signal a logic one) are considered to be the value of 00H by the BIP8 block. When 8B/10B encoding is enabled the BIP-8 value is invalid and should not be used.

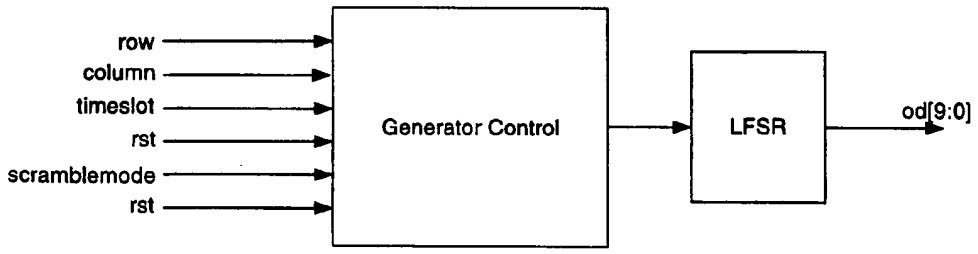
The BIP8 block inverts the BIP-8 calculated value (resulting in a B1 error on the egress data stream) when bit six of the TSEC Control and Status register (B1ERR) undergoes a logic zero to logic one transition. The B1ERR bit is cleared once the inversion of the BIP-8 calculated value has occurred at which point the B1ERR bit may be re-written with a logic one to initiate another B1 error.

Note that once enabled, B1 insertion is erred for a maximum of 250 microseconds due to the BIP-8 calculation requirements.

1.1.4.1.6 PRBS Generator Block

The PRBS Generator block implements a standard pseudo random bit sequence generation function. The block diagram for the PRBS Generator block is shown in Figure 8.

Figure 8 – PRBS Generator Block Diagram



The PRBS Generator uses a $x^{23}+x^{18}+1$ generator polynomial and must do both eight-bit and ten-bit increments in the bit sequence as the output can either be eight bits wide (SCR_EN logic 1 and EBTB_EN logic 0) or ten bits wide (EBTB_EN logic 1). The logic formulas for these sequence increments are shown in Table 2.

Table 2 – PRBS Shift Logic Formulas

Sequence Shift	Formula
10 bits	$X_N = X_{N-10}, N \geq 11$ $X_N = X_{N+13} \oplus X_{N+8}, N \leq 10$
8 bits	$X_N = X_{N-8}, N \geq 9$ $X_N = X_{N+15} \oplus X_{N+10}, N \leq 8$

The PRBS Generator must also differentiate between a raw PRBS output stream on the OD[9:0] (EBTB_EN logic 1) or OD[7:0] (EBTB_EN logic 0 and SCR_EN logic 1) data bus and PRBS within a SONET SPE, excluding path overhead and fixed stuff columns.

The PRBS Generator also inverts the PRBS (PRBS_INV, bit four of the TSEC Control and Status register, is logic high) or outputted as the normal PRBS sequence (PRBS_INV is logic low).

When PRBS is disabled the input data passes through the PRBS Generator block unaltered.

As this generator has the possibility of being stuck in the state of all zeroes in the shift register, the PRBS Generator block has a self-correction capability. When the all zero state is detected the shift register is loaded with all ones.

1.1.4.1.7 Output Select Block

The Output Select block consists of a three-to-one multiplexor. The inputs to the multiplexor are ten bits wide and initiate from the 8B/10B Encoder block, the PRBS Generator block and the TSEC Test Pattern register (normal mode register 01H). The output is placed on the OD[9:0] output. The output data, for a given input configuration may be inverted, depending on the value of OD_INV (bit five of the TSEC Control and Status register).

1.2 Receiver Definition

1.2.1 FEATURES

- Operates in either 8B/10B or scrambled NRZ (SNRZ) decode mode.
- Byte and frame aligns on 8B/10B based serial TeleCombus encoded data in 8B/10B framing mode.
- Byte and frame aligns on SNRZ encoded data in SNRZ framing mode.
- Interfaces to an analog DRU that provides 8 bit data or 10 bit data and divided by 8 or divided by 10 line rate clock for SNRZ and 8B/10B encoded data respectively.
- Operates at STS12 and STS48 input data rates.
- Optionally descrambles SNRZ data as per ANSI T1.105-1995.
- Optionally outputs 8 bit decoded 8B/10B and a status bit to identify the type of character (data or control).
- Outputs a transport frame alignment signal for both decoded 8B/10B data and descrambled SNRZ framing modes.
- Detects 8B/10B line code violations (LCVs) and accumulates them in an internal 8-bit register.
- Detects BIP8 errors and accumulates them in a 8-bit register.
- Flags line code violations in the data stream and over writes the corresponding data value with a special character that can be used by downstream blocks to identify LCV's.
- Optionally operates in bypass mode that passes input data to the output for use by PRBS diagnostics.
- Optionally operates in a diagnostic short frame mode that processes a shortened payload envelope (9 bytes per row).
- Provides a 16-bit Extended Common Bus Interface for microprocessor access to status and control registers.
- Monitors activity on the input data and input clock

1.2.2 DESCRIPTION

The PM5246 Receive Scrambled 8B/10B Encoded Data Framer (RSEF) can be combined with elements from the PMC Telecom System Block (TSB) library of ASIC functional blocks to form complete communications subsystems that can be realized on a single high-performance integrated circuit. Refer to section 1.2.4 for the RSEF block diagram.

The RSEF works in conjunction with an upstream block that packs consecutive bits from an incoming serial link into a parallel bit stream with arbitrary alignment. Data is accepted that is in either 8B/10B encoded format or as scrambled NRZ (SNRZ) data. The framing mode input (NRZMODE) selects the input data-framing format. The RSEF accepts data via a 10-bit interface (ID[9:0]). The RSEF expects a divided by eight (SNRZ framing mode) or divided by ten line rate clock (8B/10B framing mode).

In 8B/10B framing mode, the RSEF character alignment block uses the K28.5 control character (identifies the J0 byte) as a search pattern to determine correct byte alignment from the arbitrarily aligned input data. The character alignment block searches all possible input bit positions for the search pattern when in the out-of-character-alignment state. The internal byte alignment boundary is set and the character alignment block changes to the in-character-alignment state when the character alignment block finds a match to the search pattern. The character alignment block monitors line-code violations (LCV) reported by the 8B/10B decoder. The RSEF will enter the out-of-character-alignment state and begin searching for the K28.5 character if five or more line code violations are received within a 15 character window.

In SNRZ framing mode, the RSEF character alignment block transitions from the out-of-character-alignment state to the in-character alignment state when three A1 characters followed by three A2 characters are found. The character alignment block will transition from the in-character-alignment state to the out-of-character-alignment state when the frame alignment block transitions to the out-of-frame state.

The frame alignment block synchronizes internal counters to the frame timing of the input data stream. The frame alignment block outputs an out of frame (OOF) signal that is initially asserted to indicate the RSEF is not aligned to a frame. The first character alignment will result in the frame alignment block changing to the hunt state. A single correctly aligned frame must be received before the frame alignment block de-asserts the OOF signal to indicate an in-frame condition. Similarly, receiving four improperly aligned frames when in the in-frame condition will result in the OOF signal being asserted and a return to the initial out-of-frame state. The character alignment information is processed by the frame alignment block whenever the frame alignment block is in the out-of-

frame state. The framing block decodes the frame counters to generate the timing signals used by other blocks within the RSEF.

The operation of the frame alignment block for each framing mode differs in the pattern used to verify correct frame alignment has been achieved and the when the pattern is checked. The patterns used by the frame alignment block to determine correct frame alignment is the K28.5 character when in 8B/10B framing mode and the first A1 byte / first four bits received of the last A2 byte of row 1 when in SNRZ framing mode. The characters of the expected framing pattern must occur at the correct frame count for the frame alignment state machine to consider the current frame alignment as valid. The A1 / A2 pattern is checked every frame when in SNRZ framing mode. The K28.5 character is checked only when it occurs in the bit stream when in 8B/10B framing mode. The modulus of the frame counter is configurable via the STSNMODE pin to support STS12, STS48, and STS51 line coding.

Once the data has been character aligned to determine a byte boundary and frame aligned to synchronize internal counters to the frame timing, the data path can be configured to process the data via the descrambler enable (DESCREN) and 8b/10b decode enable signals (EBTBEN). Viable processing options are to 8B/10B decode the data, descramble the data and 8B/10B decode then descramble the data. The processed input data is output on OD[7:0].

The RSEF outputs 10 bit data (OD[9:0]), the OJ0 pulse to identify the output byte as occurring at the J0 byte position, and a K signal to indicate the received data is a control character or line code violation when 8B/10B decoding is enabled. The character alignment, frame alignment, 8b/10b decoder and descrambler are bypassed when the BYPASS control bit of the ECBI is set high. In bypass mode, the input data bus (ID[9:0]) that has optionally been inverted via the ECBI PININV bit is passed to the output data bus (OD[9:0]). The K and OJ0 signals are low in bypass mode. Bypass mode allows for raw data to be passed through the RSEF for system level link testing. The two most significant bits of the output data bus are not valid when not in bypass mode and either of the descrambler or 8b/10b decoder are enabled.

The 8B/10B decoder decodes according to the IEEE802.3 standard with the exception of special characters used to encode TeleCombus control signals (the line codes for positive and negative accumulated disparity of the control characters K28.0, K28.4, K38.7, K23.7, K27.7, K29.7 and K30.7). For these special control codes, the RSEF substitutes the decoded data with an arbitrarily defined value (see Table 13) and sets the K control signal high. For all other characters, with the exception of characters that violate the 8B/10B line code, the RSEF outputs the decoded 8B/10B data value as defined by the IEEE802.3 specification and sets K low.

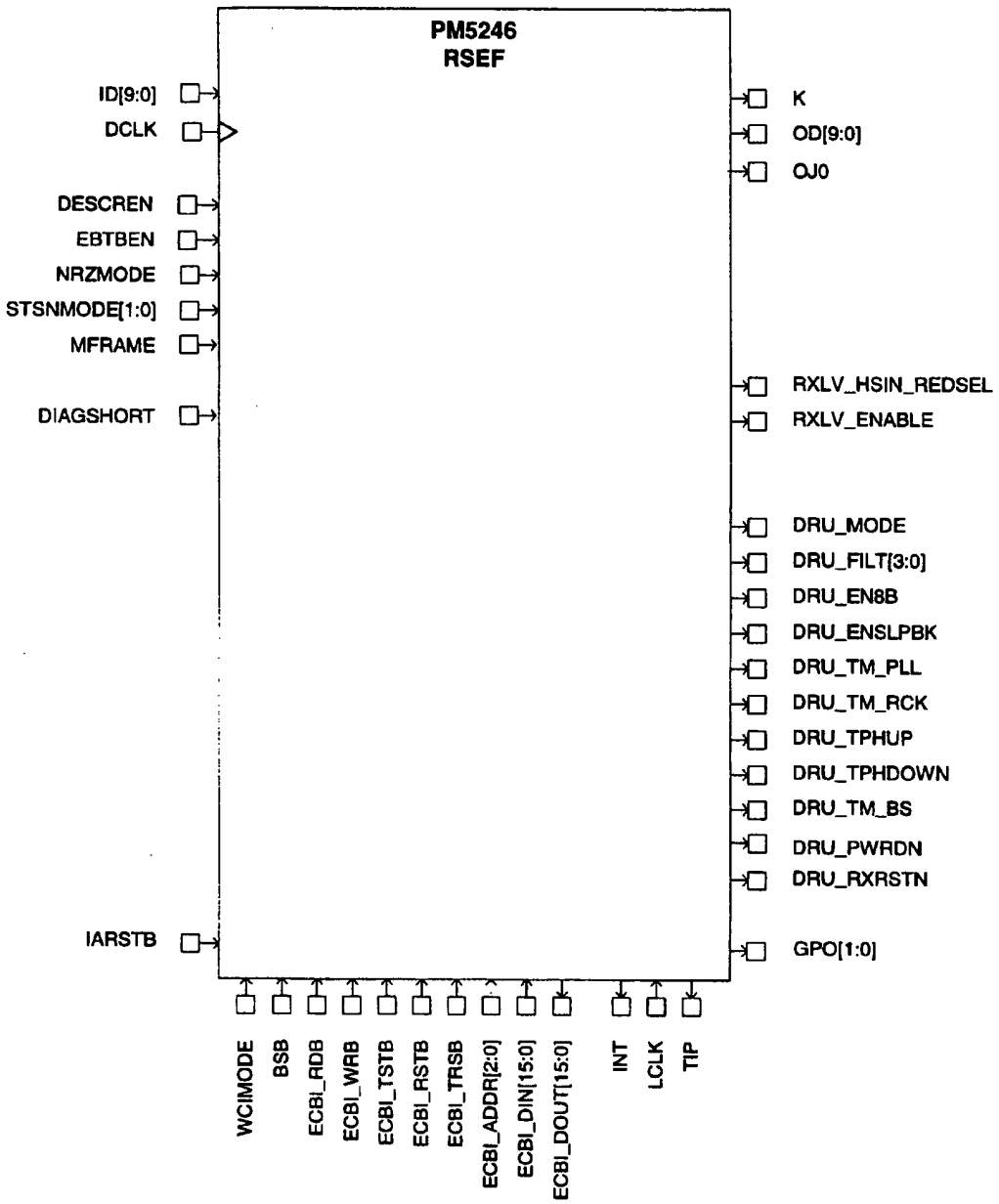
The 8B/10B decoder monitors and reports line code violations. A LCV occurs when the incoming data is not one of the valid 8B/10B characters or when the

running disparity is incorrect. The TeleCombus special characters K28.0-, K28.0+, K28.4-, K27.7-, K27.7+, K28.7-, K28.7+, K29.7-, K29.7+, K30.7-, K30.7+, and K28.4+ are not considered LCVs but do effect current running disparity. A maskable interrupt is generated when a line code violation occurs. The 8B/10B decoder maintains a 8-bit count of the total number of line code violations when the 8b/10b decoder is enabled. To allow downstream blocks an opportunity to properly deal with LCV's, characters that have a LCV are substituted with h0F and K is asserted high.

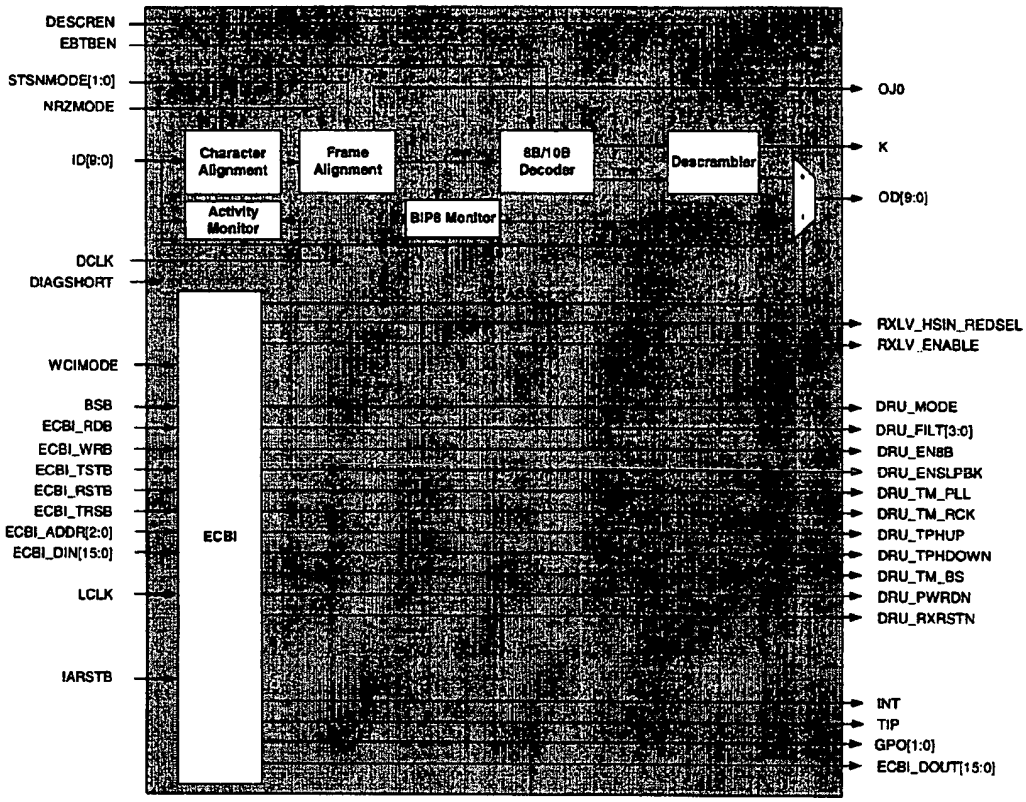
The descrambler block descrambles data according to the SONET/SDH specifications with the exception of control characters output from 8b/10b decoder (K is high), which are not descrambled. The descrambler does not modify the K signal output from the 8b/10b decoder. The descrambler passes output data from the 8b/10b decoder without modification when DESCREN is asserted low.

The BIP8 block calculates the BIP8 parity over a received frame of scrambled data and compares the result to the expected value in the B1 byte position of the next descrambled frame. The BIP8 errors are counted when 8b/10b decoding is disabled. A maskable interrupt is generated when a BIP8 violation is detected. The BIP8 calculation is intended to be used for monitoring SNRZ coded line data. The BIP8 interrupt should be masked and LCV's monitored for 8b/10b encoded data or for 8b./10b encoded SNRZ data.

1.2.3 ICON


PM5246
RSEF
ID[9:0]
DCLK
DESCREN
EBTBEN
NRZMODE
STSNMODE[1:0]
MFRAME
DIAGSHORT
IARSTB
K
OD[9:0]
OJ0
RXLV_HSIN_REDSEL
RXLV_ENABLE
DRU_MODE
DRU_FILT[3:0]
DRU_EN8B
DRU_ENSLPBK
DRU_TM_PLL
DRU_TM_RCK
DRU_TPHUP
DRU_TPHDOWN
DRU_TM_BS
DRU_PWRDN
DRU_RXRSTN
GPO[1:0]
WCIMODE
BSB
ECBI_RDB
ECBI_WRB
ECBI_TSTB
ECBI_RSTB
ECBI_TRSB
ECBI_ADDR[2:0]
ECBI_DIN[15:0]
ECBI_DOUT[15:0]
INT
LCLK
TIP

1.2.4 BLOCK DIAGRAM



1.2.5 FUNCTIONAL DESCRIPTION

1.2.5.1 Modes

There are six mode pins on the RSEF: STSNMODE, NRZMODE, EBTBEN, DESCREN, WCIMODE and DIAGSHORT. The STSNMODE, NRZMODE, WCIMODE, and DIAGSHORT modes are independent. The EBTBEN and DESCREN modes define the processing that is performed on the input data after character and frame alignment.

In addition to the modes defined by the mode pins, the PININV bit controls the polarity of the input data. The BYPASS bit of the ECBI status and control register allows data to be passed from the RSEF input data bus to the output data bus after optional polarity inversion.

1.2.5.1.1 STSNMODE

The STSNMODE pins configure the RSEF to process STS12/STM4, STS48/STM16, or STS51 frames. Refer to Table 3 for the input frequency requirement and settings for STSNMODE.

Table 3 - STSNMODE Settings

STSNMODE[1:0]	RSEF Operating Mode	DCLK Frequency
'b00	STS12/STM4	77.76 MHz
'b01	STS48/STM16	311.04 MHz
'b10	STS51	330.48 MHz

Note: RSEF Input data is passed to the output after optional polarity inversion and the STSNMODE setting is irrelevant when BYPASS mode is enabled.

1.2.5.1.2 NRZMODE

The NRZ mode signal (NRZMODE) defines the bit width of the input data, specifies the search pattern used for character alignment, and determines the algorithm used for frame alignment. Refer to Table 4 for a summary of the NRZMODE settings.

Setting NRZMODE high configures the RSEF to character and frame align to scrambled NRZ data. The RSEF processes 8 bit input data arriving on ID[7:0], DCLK is expected to be the recovered line clock divided by 8, the character alignment block searches for three A1 characters followed by three A2 characters, and the frame alignment block aligns to the first A1 character and first four bits received of the last A2 character of a STSN signal.

Setting NRZMODE low configures the RSEF to character and frame align to 8B/10B encoded data. The RSEF processes 10 bit input data arriving on ID[9:0], DCLK is expected to be the recovered line clock divided by 10, and both the character and frame alignment blocks align to the K28.5+ or K28.5- control character.

Table 4 - NRZMODE Settings

NRZMODE / DESC.	INPUT DATA	DCLK	Character Alignment Searches For:	Frame Alignment Error Check:
'b0 Align to 8b/10b TeleCombus Encoded Data	ID[9:0]	Line clock divided by 10	K28.5 (J0)	Every received K28.5 character arrives at the correct position

'b1 Align to SNRZ Data	ID[7:0]	Line clock divided by 8	3 A1's followed by 3 A2's	The first A1 and first four bits received last A2 (bits 7 – 4) of row 1 at the correct position every frame

Note: RSEF Input data is passed directly to the output and the NRZMODE setting is irrelevant when BYPASS mode is enabled.

1.2.5.1.3 EBTBEN and DESCREN Modes

Setting the 8B/10B enable (EBTBEN) signal high enables the 8B/10B decoder. The 8B/10B decoder performs 8B/10B character decoding according to the IEEE802.3 standard with the exception of special characters used to encode TeleCombus control signals. The line codes for positive and negative accumulated disparity of the control characters K28.0, K28.4, K38.7, K23.7, K27.7, K29.7 and K30.7 are allocated for this purpose. The 8B/10B decoder marks the decoded data as a control character or line code violation using the K signal. When K is high, the value of OD identifies the type of character received, either a control character or a character with a line code violation, as specified in Table 13. The K output of the 8B/10B decoder is always low and data is passed from the input of the 8b/10b decoder to the output of the 8b/10b decoder without modification when EBTBEN is low.

Setting the descrambler enable (DESCREN) signal high enables the descrambler. The descrambler operates as specified by the ANSI-T1.105-1995 and ITU G.707 standards. The K output of the 8b/10b decoder is passed unaltered by the descrambler. The descrambler does not descramble data arriving at its input when K is high. When DESCREN is low, the descrambler passes K and data from the input of the descrambler to the output of the descrambler without modification (the descrambler is disabled).

The EBTBEN and DESCREN input signals define the decoding operations performed on the input data after the RSEF has completed character and frame alignment. Data is processed first by the 8B/10B decoder and then by the descrambler. Refer to Table 5 for a summary of the data processing modes provided by the RSEF.

Table 5 -EBTBEN / DESCREN Mode Settings

EBTBEN	DESCREN	OD	K	OJ0
0	0	Not supported	Undefined	Undefined
0	1	OD[7:0] is Descrambled ID	Always 0	High when OD is the J0 Byte
1	0	OD[7:0] is 8B/10B Decoded ID	0	High when OD is the J0 Byte
		OD[7:0] is Control or LCV indication as per Table 13	1	High when OD is the J0 Byte

1	1	OD[7:0] is 8B/10B Decoded then descrambled ID	0	High when OD is the J0 Byte
		OD[7:0] is Control or LCV indication as per Table 13	1	High when OD is the J0 Byte

Note: RSEF Input data is passed to the output after optional polarity inverstion and the EBTBEN / DESCREN settings are irrelevant when BYPASS mode is enabled.

1.2.5.1.4 BYPASS Mode

When the BYPASS bit of the ECBI status and control register is high, the input data present on ID[9:0] is passed to OD[9:0] after optional polarity inversion.

When the BYPASS bit of the ECBI status and control register is low, the input data is character aligned, frame aligned, and decoded as specified by the NRZMODE, EBTBEN, and DESCREN mode pin settings.

1.2.5.1.5 PININV Mode

When the PININV bit of the ECBI status and control register is high, the input data present on ID[9:0] is inverted prior to processing by the RSEF. When the PININV bit is low, the input data is not inverted prior to processing by the RSEF.

1.2.5.1.6 WCIMODE

When WCIMODE is high, interrupts are cleared on a write of logic 1 to the corresponding status bit. When WCIMODE is low, interrupts are cleared on a read of the interrupt status register.

Table 6 - WCIMODE Settings

WCIMODE	RSEF Operating Mode
'b0	Interrupts are cleared on a read of the interrupt status register
'b1	An interrupt is cleared on a write of logic 1 to the corresponding status bit.

1.2.5.1.7 DIAGSHORT

When DIAGSHORT is high, the RSEF frame alignment block operates based on a short STS1 frame that consists of 3 overhead and 9 payload bytes per row. Therefore, the RSEF processes shortened STS12/STM4, STS48/STM16, and STS51 frames based on 108, 432, and 459 bytes of payload data per row

respectively when in DIAGSHORT mode. The DIAGSHORT mode is intended to reduce the amount of time required to simulate a large number of frames or for production testing. DIAGSHORT must be set low for normal operation.

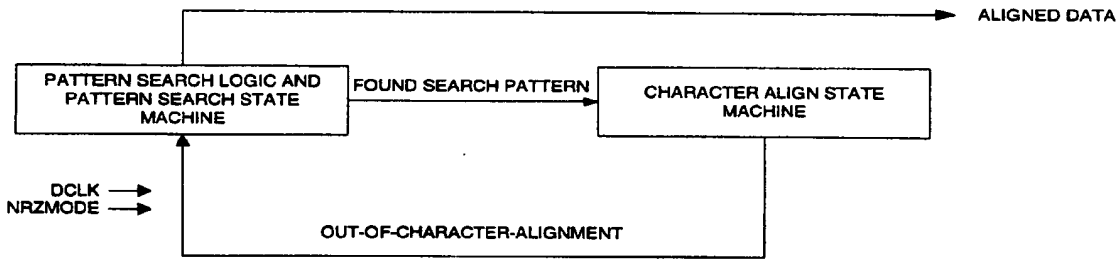
Table 7 - DIAGSHORT Settings

DIAGSHORT	RSEF Operating Mode
'b0	The RSEF processes frames with a 87 byte STS1 payload
'b1	The RSEF processes frames with a 9-byte STS1 payload.

1.2.5.2 Character Alignment

The character alignment block (Figure 9) consists of pattern search logic and a character alignment state machine. The search logic locates character boundaries in the arbitrarily aligned incoming 8B/10B or SNRZ data stream and provides an indication to the character alignment state machine (found-search-pattern) when a character boundary has been found. The character alignment state machine maintains two states: out-of-character-alignment and in-character-alignment.

Figure 9 - Character Alignment Block



1.2.5.2.1 Pattern Search Logic and Pattern Search State Machine

The pattern search logic looks for a single character while the pattern search state machine looks for a sequence of such characters. In 8B/10B framing mode, the pattern search logic looks for the J0 character (K28.5) and the pattern search state machine counts a single occurrence. In SNRZ framing mode, the pattern search logic looks for A1 and A2 characters while the state machine looks for a sequence of three A1's followed by three A2's. The pattern search logic changes the character alignment of the input data once the pattern search state machine reports the correct sequence of search characters has been found and if the

character alignment state machine is in the out-of-character-alignment state. Table 8 summarizes the search patterns used in each framing mode.

Table 8 - Character Alignment Search Patterns

NRZMODE	Search Characters	Value
b0 = 8B/10B	K28.5+	b1100000101
	K28.5-	b0011111010
b1 = SNRZ	3 A1's followed by 3 A2's	hF6F6F6282828

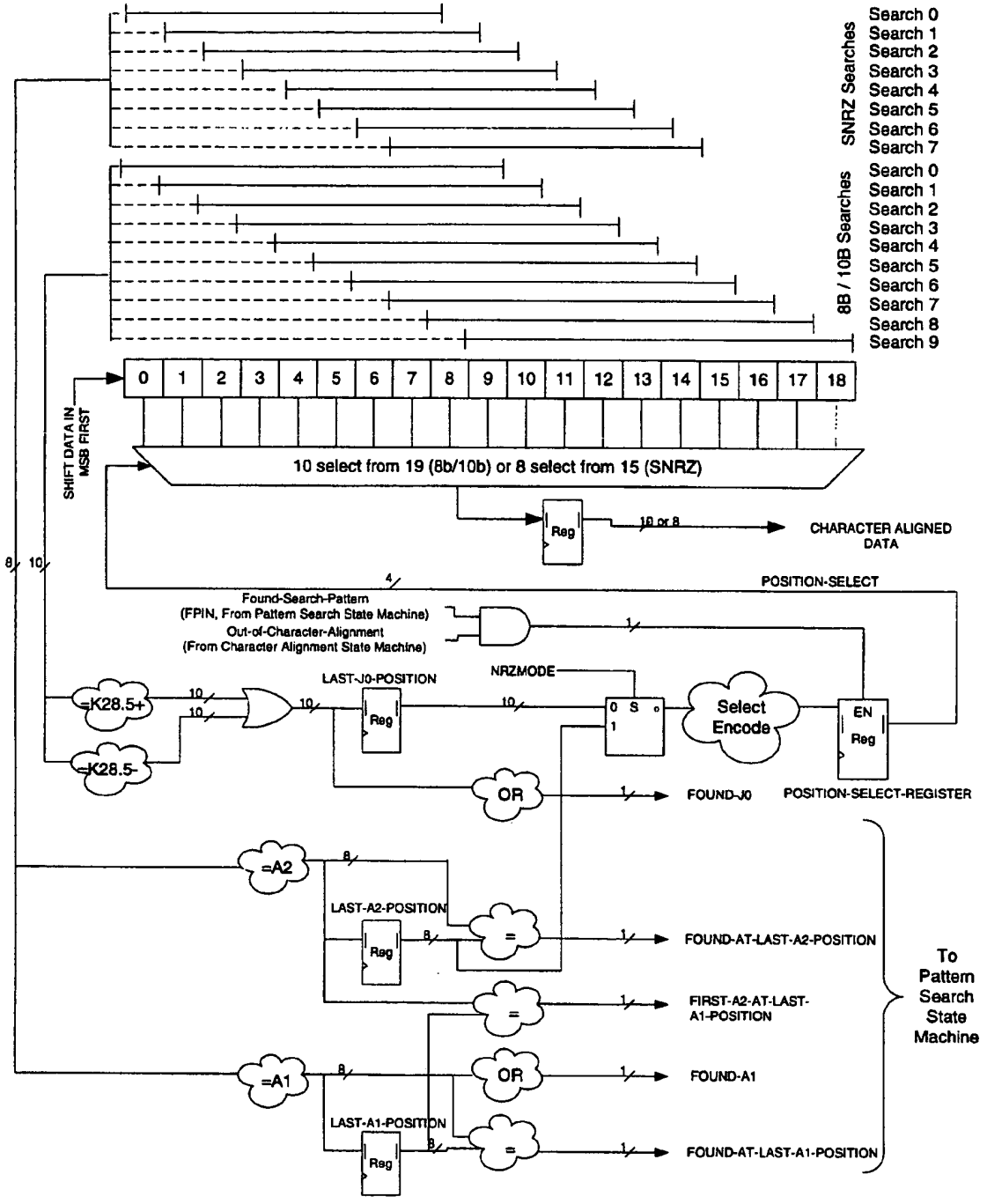
The method used to find the search pattern by the pattern search logic involves looking for all possible positions of the search pattern character (A1, A2, K28.5+, or K28.5-) over two consecutively received characters as shown in Figure 10. In 8B/10B framing mode, the search character consists of the 10-bit K28.5+ or K28.5- character. The search is completed over 19 bits with a match occurring in one of the 10 possible character alignments. In scrambled NRZ framing mode, 8 bit data is received from the ID[7:0] input bus. The search is completed over 15 bits with one or more matches occurring in 8 possible character alignments.

The pattern search logic computes the logic signals required by the pattern search state machine to determine if a single J0 character (8b/10b framing mode) or three A1 characters followed by three A2 characters (SNRZ framing mode) have been found. The signals found-J0 and found-A1 indicate if J0 or A1 characters respectively have been found in any of the possible search positions. The signals found-at-last-a1-position and found-at-last-a2-position are used by the pattern search state machine to ensure successive A1 and A2 characters are found at the same alignment position. The signal first-a2-at-last-a1-position is asserted if an A2 character is found immediately after and at the same position as an A1 character.

The state machine to detect 3 consecutive A1 characters followed by three consecutive A2 characters or a single J0 character is shown in Figure 11. The state machine starts in the search state and transitions to the found-1-a1 state when the RSEF is in SNRZ framing mode and the pattern search logic has found an A1 character. The state machine will progress through similar states upon receiving two more A1 characters at the same position as the first received A1 character until the found-3-a1 state is reached. The A1 characters must occur consecutively. The state machine will wait at the found-3-a1 state as long as consecutive A1 characters are found at the same position. If an A1 character is followed by an A2 character at the same position, the state machine will progress to the found-3-a1-and-1-a2 state. Two more A2 characters at the same position will result in the state machine reaching the found-3-A1-and-3-A2-or-J0 state. The A2 characters must occur consecutively. The "found search pattern" signal, also used as the frame pulse in signal for the frame alignment block, is asserted when in the found-3-A1-and-3-A2-or-J0 state. The state machine transitions back

to the search state if a valid A1 or A2 character is not received at the same position as the last character received and if a A1 character is not found. Similarly, the state machine transitions back to the found-1-A1 state if a valid A1 or A2 character is not received at the same position as the last character received but an A1 character is found. If the RSEF is in 8B/10B framing mode and a valid J0 character is received, the state machine transitions directly to the found-3-A1-and-3-A2-or-1J0 state.

Figure 10 - Character Alignment Pattern Search Logic

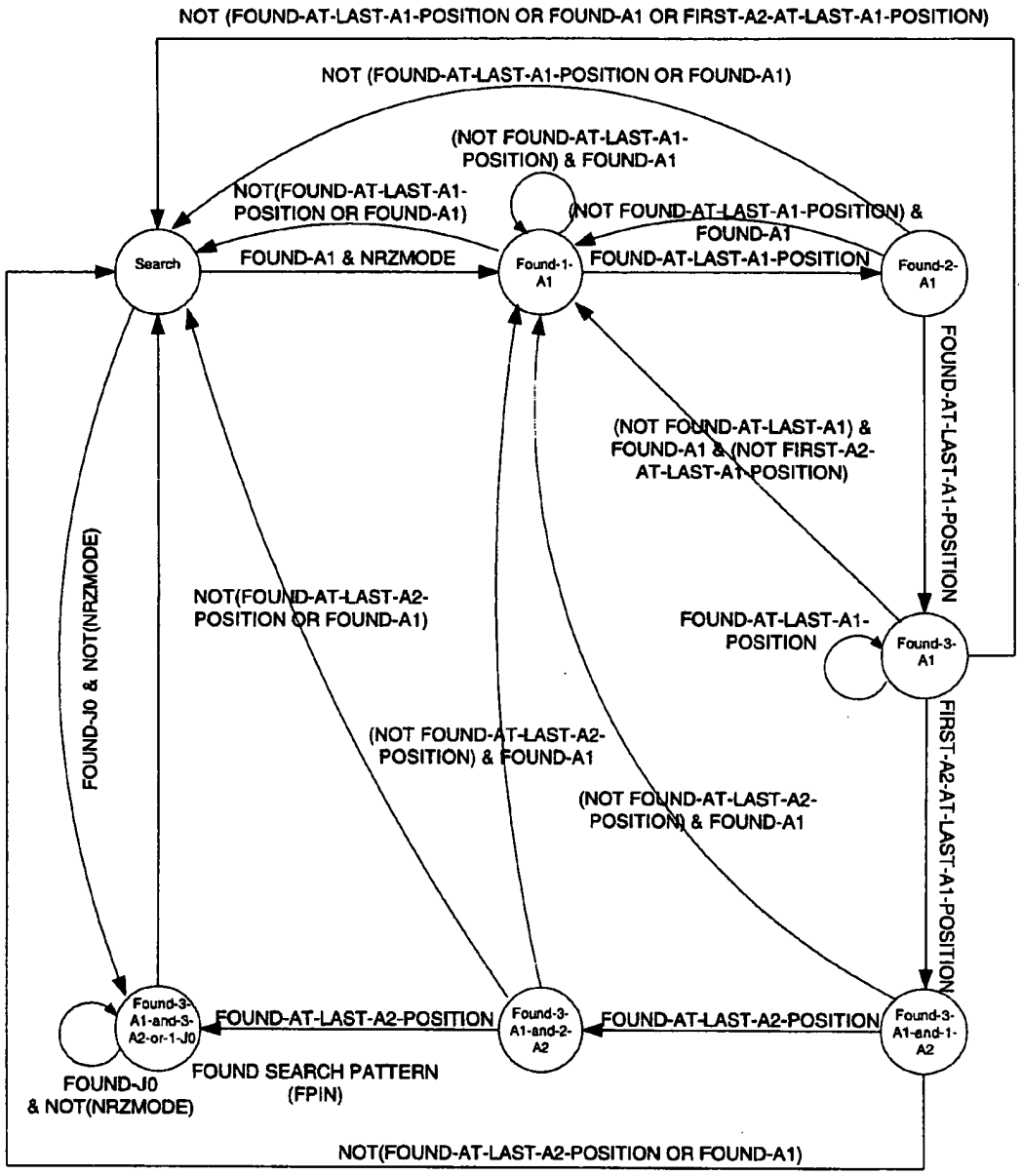


The result of the parallel search produces a signal (found-search-pattern) to the character alignment state machine to indicate that a search pattern has been

found. The same signal is used as a framing pulse for the frame alignment block (FPIN). A new position-select-register value is loaded from the last-a2-position register (SNRZ framing mode) or last-j0-position register (8b/10b framing mode) when the search pattern is found and the character alignment state machine is in the out-of-character-alignment state. Correctly aligned data is subsequently forwarded by selecting bits from the shift register as specified by the position-select-register.

Note: FPIN occurs one clock cycle after the corresponding data (either third A2 or J0) has been output from the character alignment block.

Figure 11 - Pattern Search State Machine

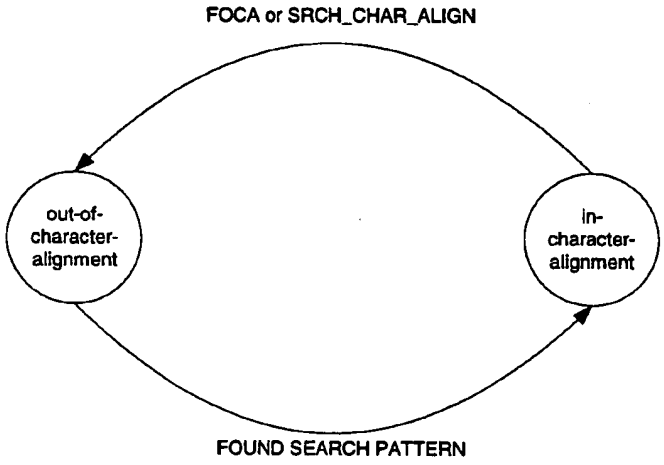


Note: The pattern search state machine will transition to the search state from the Found-1-A1, Found-2-A1, Found-3-A1 or Found-3-A1-and-1-A2 if 8b/10b framing is enabled (NRZMODE is low). The condition for this transistion is not shown in Figure 11.

1.2.5.2.2 Character Alignment State Machine – SNRZ Framing Mode

The character alignment state machine operates as shown in Figure 12 when in SNRZ framing mode. The character alignment state machine may be in one of two states, the in-character-alignment state and the out-of-character alignment state. The state machine is initially in the out-of-character-alignment state and changes to the in-character-alignment state when the pattern search state machine indicates a search pattern has been found. The state machine changes to the out-of-character-alignment state if the frame alignment block is in the out-of-frame state (SRCH_CHAR_ALIGN is asserted high) or if a zero to one transition is detected on the force out of character alignment (FOCA) status bit. The character alignment block continuously looks for a valid search pattern but only updates the character alignment if the character alignment state machine is in the out-of-character-alignment state and a valid search pattern has been found.

Figure 12 - Character Alignment State Machine - SNRZ Framing Mode

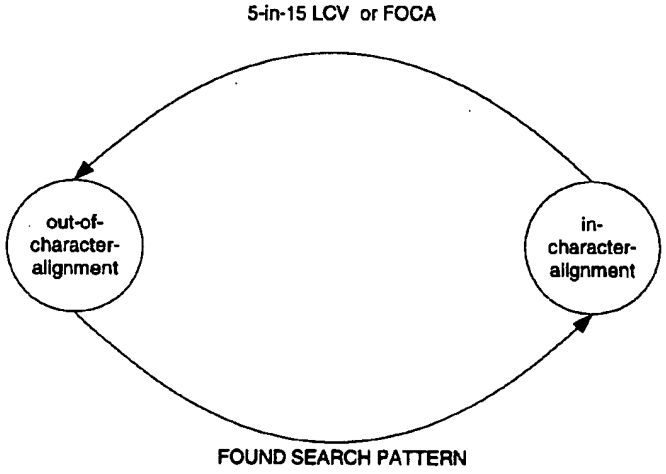


1.2.5.2.3 Character Alignment State Machine – 8B/10B Framing Mode

The character alignment state machine operates as shown in Figure 13 when in 8B/10B framing mode. The character alignment state machine transitions from the out-of-character alignment state to the in-character-alignment state when the pattern search state machine indicates the search pattern (J0 character) has been found. The character alignment state machine monitors the number of line code violations (LCV) reported by the 8B/10B decoder to determine if a transition to the out-of-character alignment state is necessary. An LCV occurs when the incoming data is not one of the valid 8B/10B characters or when the running disparity is incorrect. The TeleCombus special characters (K28.0-, K28.0+, K28.4-, K27.7-, K27.7+, K28.7-, K28.7+, K29.7-, K29.7+ K30.7-, K30.7+, and

K28.4+) are ignored for LCV purposes. If 5 or more LCVs are detected within a 15-character window the character alignment block will change to the out-of-character-alignment state. Upon return to the in-character-alignment state the 5 in 15 LCV count is cleared.

Figure 13 - Character Alignment State Machine – 8B/10B Framing Mode

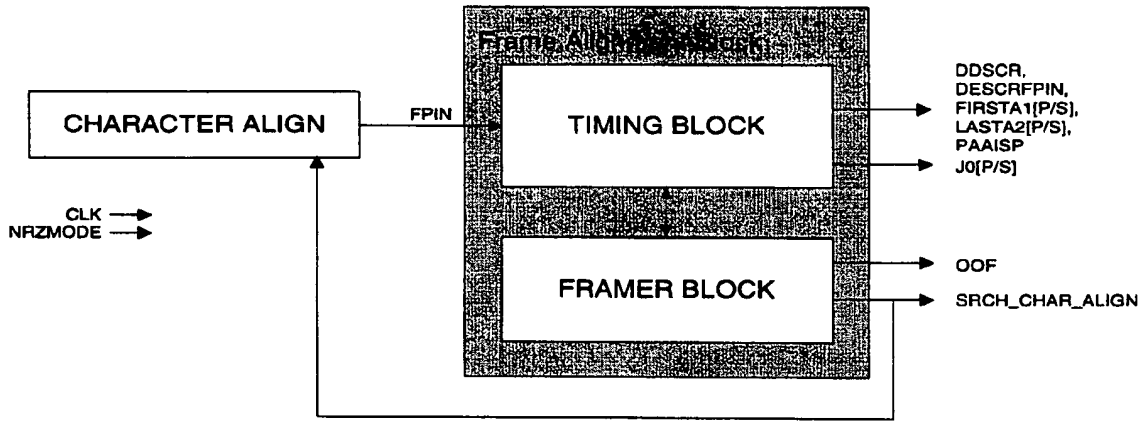


Note: The character alignment state machine may be set to the out-of-character-alignment state by creating a zero to one transition on the ECBI signal FOCA (Force out-of-character-alignment) when in either SNRZ or 8B/10B framing mode.

1.2.5.3 Frame Alignment Block

The function of the frame alignment block is two fold: to generate the timing required to identify the incoming bytes as required by the other blocks within the RSEF and to monitor the bytes received from the character alignment block to detect an out of frame condition. The two functions are implemented in timing and framing sub-blocks respectively. Refer to Figure 14 for a simplified block diagram showing the control relationship between the character alignment block and frame alignment sub-blocks.

Figure 14 - Frame Alignment / Character Alignment Connectivity



1.2.5.3.1 Timing Block

The timing block maintains counters of the current row, column, and STS signal of the received characters. The counter values are decoded by logic within the timing block to generate controls signals that are used by the framer block and the descrambler.

The timing block expects 9 rows of 90 columns per STS-1/STM-0. The RSEF block supports framing of up to an STS-51 signal. Three counters are needed: one row counter (1h to 9h), one column counter (1h to 5Ah) and one STS counter (1h to 33h). The STS counter is incremented every DCLK cycle. The column counter is incremented when the STS counter wraps from its terminal value to 1, and the row counter is incremented when the column counter wraps from its terminal value to 1. The terminal counts of the four counters are summarised in for both normal (DIAGSHORT is set low) and diagnostic short modes (DIAGSHORT is set high). The reset value for all counters h1.

Table 9 - Decimal/Hex Terminal Counts for the Timing Block Counters

Counter	Normal Terminal Count	Diagnostic Short Terminal Count
STS	12 = h0c When STSNMODE = b00 48 = h30 When STSNMODE = b01 51 = h33 When STSNMODE = b10	As per normal terminal count
Column	90 = h5a	12 = h0c
Row	9 = h9	As per normal terminal Count

Two sets of counters are used, a primary row, column and STS-1/STM-0 counter and a secondary row, column and STS-1/STM-0 counters. When the framer block is in the in-frame-alignment state, the primary and secondary counters are free running on the last frame alignment (the input frame pulse is ignored). When the framer block is in the hunt state, the primary counters keep free running on the last frame alignment and the secondary counters start looking for a new frame alignment based on the input frame pulse. The secondary and primary counters are reset as shown in Table 10 when the framer block transitions from the hunt state to the in-frame-alignment state. Note that the input frame pulse marks different byte locations within an STS frame depending on the framing mode.

Table 10 - Framing Pulse Position

	SNRZ Framing	**8B/10B Framing**
FPIN Marks Byte 1 Bytes After:	Third A2 Byte	J0 Byte
Secondary STS Count Reset To:	3	1
Secondary Column Count Reset To:	2	3
Secondary Row Count Reset To:	1	1

The timing block decodes the primary and secondary counters to mark the position of the J0 byte. J0P indicates the primary counter is at the J0 byte position within a frame. J0S indicates the secondary counter is at the J0 byte position within a frame.

The signals required to reset and enable the descrambler are generated within the timing block. The disable descrambler signal (DDSCR) is asserted high when the primary row, column, and STS counters are at the A1, A2, section trace (J0) or Section Growth (Z0) byte positions. The DDSCR signal is used to disable the descrambler. The descrambler frame pulse in (DESCFPIN) signal is asserted high when the primary counter is at the Z0 byte position of the last STSN signal in a frame and is used to reset the descrambler. The DESCR_B1 signal is asserted high when the descrambler output is the B1 byte and is used by the BIP8 block.

The timing block generates signals used by the framer block to check the validity of the first A1 character and first four bits received of the last A2 character. The FIRSTA1S and FIRSTA1P signals are asserted high when the secondary and primary counts respectively are at the first A1 character position within a frame. The LASTA2S and LASTA2P signals are asserted high when the secondary and primary counts respectively are at the last A2 character position. Table 11 summarises the counts that are decoded to create the control signals generated by the timing block.

The PAAISP control signal is used to identify bytes for substitution with internal code h01 (K asserted high) when the PAAIS bit of the ECBI status register is set and the

frame alignment block is in the out-of-frame-alignment state. The PAAIS bit marks the H1, H2, H3 bytes and all bytes within the SPE.

Table 11 - Decoded Counts for Timing Block Control Signals

Control Signal	Count when Asserted	Purpose
J0[P/S]	STS = 1 COL = 3 ROW = 1	Marks the J0 Byte, frame alignment position indication to downstream blocks
DDSCR	Primary Counter: STS = All COL is <= 3 and >= 1 ROW = 1	Disable descrambling for the transport overhead bytes A1, A2, J0, and Z0. Decode primary counter only.
DESCRFPIN	Primary Counter: STS = 12 When STSNMODE = b00 STS = 48 When STSNMODE = b01 STS = 51 When STSNMODE = b10 ROW = 1 COL = 3	Reset the descrambler at the Z0 byte of the last STS-N signal of Row 1. Decode primary counter only.
DESCRB1	Primary Counter: STS = 1 COL = 1 ROW = 2	Marks the B1 byte. Used by the the BIP8 circuitry to extract the B1 byte from the descrambled data
FIRSTA1[P/S]	STS = 1 COL = 1 ROW = 1	Marks the first A1 byte of Row 1, used by the frame alignment state machine
LASTA2[P/S]	STS = 12 When STSNMODE = b00 STS = 48 When STSNMODE = b01 STS = 51 When STSNMODE = b10 COL = 2 ROW = 1	Marks the last A2 byte of Row 1, used by the frame alignment state machine
PAAISP	(ROW = 4, COL is <=3 and >=1, STS= ALL) OR (ROW = ALL, COL is <=90 and >=4, STS=ALL)	Marks the H1, H2, H3 and SPE for all STS signals. Decode primary counter only.

1.2.5.3.2 Framer Block

The framer block locates frame boundaries in the incoming data stream and detects an out of frame condition by checking the data stream against an expected framing

pattern. The monitored framing pattern and the method used to check the framing pattern is dependent on the framing mode as defined in Table 12. In SNRZ framing mode, a framing error occurs when an A1 or first four bits received of an A2 character is not found when the primary frame counters are at the appropriate count. In 8b/10b framing mode, a framing error occurs when a K28.5 character is received and the primary frame counter does not indicate the J0 byte position.

Note: The frame block checks any received K28.5 (J0) characters against the primary count when in 8B/10B framing mode, and therefore, does not require a K28.5 character to occur at the J0 position for every frame received.

Table 12 - Overhead Bytes Used for the Framing Pattern

NRZMODE	Overhead Bytes	Binary Values	Frame Alignment Error Check
SNRZ	First A1 and Last A2 byte of row 0	A1: hF6 A2: h28	Check for A1 / A2 input data at the first frame of every multi-frame
8B/10B	K28.5+ or K28.5-	K28.5+: b1100000101 K28.5-: b0011111010	Check the frame count for every K28.5 character received

1.2.5.3.3 Frame Alignment State Machine for SNRZ Framing Mode

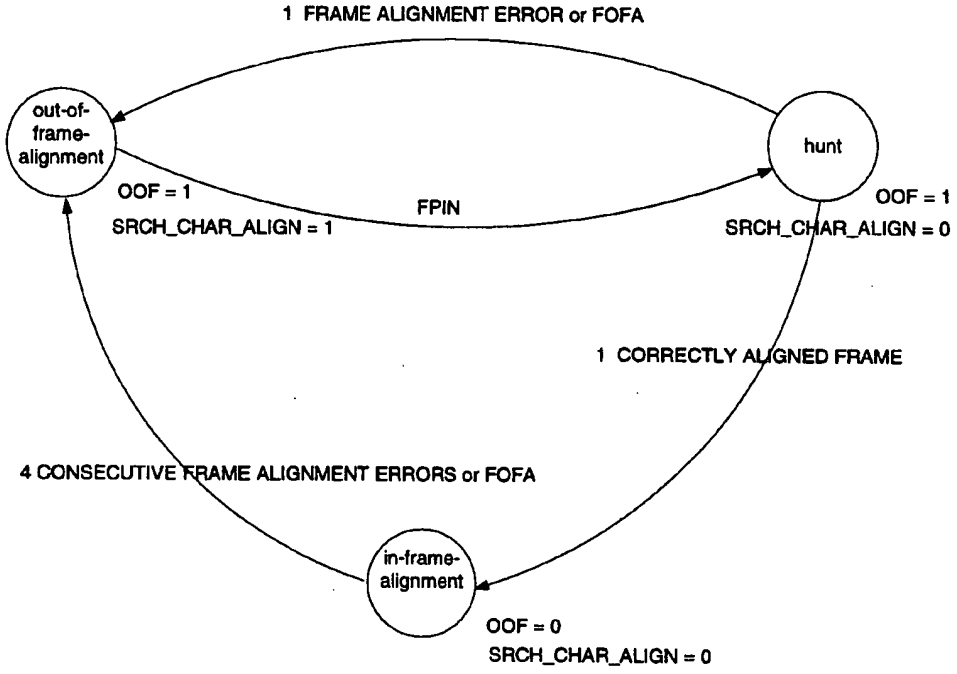
The framer block state diagram is shown in Figure 15 for SNRZ framing mode. The frame alignment state machine may be in one of three states, out-of-frame-alignment, hunt and in-frame-alignment.

When the state machine is in the out-of-frame-alignment state, the primary frame counters maintain the current frame alignment and the frame alignment state machine waits for the frame pulse signal from the character alignment block. One occurrence of the frame pulse causes the frame alignment state machine to transition to the hunt state and the secondary frame counters to be initialized to the appropriate count. One occurrence of the framing pattern (first A1 / first four bits received of the last A2) in the immediately following frame at the position indicated by the secondary counters will result in the state machine changing to the in-frame-alignment state. The primary counters are set to the value of the secondary counters when in the in-frame-alignment state. Any single frame alignment error while in the hunt state will result in a change to the out-of-frame-alignment state. Four consecutive frame alignment errors while in the in-frame-alignment state will result in a change to the out-of-frame-alignment state. The srch__char_align output is used by the character alignment block to enable updates of the character alignment whenever the frame alignment state machine is in the out-of-frame state.

Note: Two frames must always be received to transition from the out-of-frame-alignment state to the in-frame-alignment state. The first frame sets the frame timing counters and results in a transition from the out-of-frame-alignment state to the hunt state. The second frame must occur with the same frame alignment as the first to transition from the hunt state to the in-frame-alignment state.

Note: The frame alignment state machine may be set to the out-of-frame-alignment state by creating a zero to one transition of the ECBI status and control register bit FOFA (Force out-of-frame-alignment).

Figure 15 - Framer Block State Diagram for SNRZ Framing Mode



Note: The frame alignment state machine ignores the frame pulse from the character alignment block when in any state except the out-of-frame-alignment state.

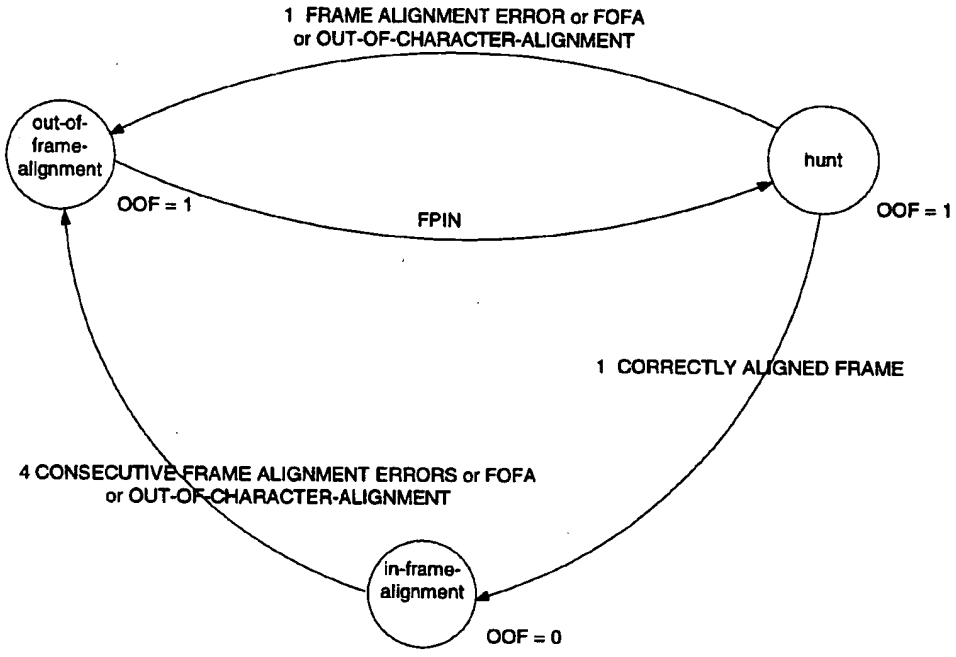
1.2.5.3.4 Frame Alignment State Machine for 8B/10B Framing Mode

The framer block state diagram is shown in Figure 16 for 8B/10B framing mode. The frame alignment state machine may be in one of three states, out-of-frame-alignment, hunt and in-frame-alignment.

When the state machine is in the out-of-frame-alignment state, the primary frame counters maintain the current frame alignment and the frame alignment state machine waits for the frame pulse signal from the character alignment block. One occurrence of the frame pulse causes the frame alignment state machine to transition to the hunt state and the secondary frame counters to be initialized to the appropriate count. One occurrence of the framing pattern (K28.5) at the position indicated by the secondary counter will result in the state machine changing to the in-frame-alignment state. The primary counters are set to the value of the secondary counters when in the in-frame-alignment state. Any single frame alignment error while in the hunt state will result in a change to the out-of-frame-alignment state. Note that any number of frames that do not contain a K28.5 character may be received while in the hunt state. The first frame received with a K28.5 character will be checked for a frame alignment error.

Four consecutively received K28.5 characters at the incorrect position as predicted by the primary frame counters while in the in-frame-alignment state will result in a change to the out-of-frame-alignment state. In addition, the frame alignment state machine will transition from the hunt or in-frame-alignment states to the out-of-frame-alignment state whenever the character alignment state machine is in the out-of-character-alignment state.

Figure 16 - Framer State Diagram for 8B/10B Framing Mode



Note: The frame alignment state machine ignores the frame pulse from the character alignment block when in any state except the out-of-frame-alignment state.

Note: The frame alignment state machine may be set to the out-of-frame-alignment state by creating a zero to one transition of the ECBI status and control register bit FOFA (Force out-of-frame-alignment).

Note: Two J0 characters must be received to transition from the out-of-frame alignment state to the in-frame-alignment state with the second character occurring at the position predicted by the first.

1.2.5.4 8B/10B Decoder

The 8B/10B Decoder block decodes the incoming data according to the IEEE standard. If the received 8B/10B encoded byte is one of the serial 8B/10B TeleCombus special characters or generates a line code violation then the 8B/10B decoder asserts K high and substitutes the decoded value with the value specified in Table 13. If the received 8B/10B encoded byte is not one of the TeleCombus special characters or does not generate a line code violation, then the decoded value is placed on the output of the 8B/10B decoder and K is asserted low.

Table 13 - RSEF Output Data for TeleCombus Ctrl Characters and LCVs

Code Group Name	Curr. RD- abcdei fghj	Curr. RD+ abcdei fghj	Substituted Value / Purpose / Equivalent Parallel TeleCombus control bytes and signals
Multiplex Section Termination (MST) Level Control Characters			
K28.5	b001111 1010	b110000 0101	b00000000 Transport Frame Alignment IJ0 = 1 and IPL = 0
K28.4-	b001111 0010	-	b00000001 High-order path AIS IPAIS = 1
High-Order Path Termination (HPT) Level Control Characters			
K28.6	b001111 0110	b110000 1001	b00000010 High-order path frame alignment IJ1 = 1 and IPL = 1
K28.0-	b001111 0100	-	b00000011

			High-order path H3 byte, no negative justification event ID = H3 position and IPL = 0
K28.0+	-	b110000 1011	b00000100 High-order path PSO byte, positive justification event ID = PS0 position and IPL = 0
Low-Order Path Termination (LPT) Level Control Characters			
K28.4+	-	b110000 1101	b00000101 Low-order path AIS IVAIS = 1
K27.7-	b110110 1000	-	b00000110 Low order path frame alignment IV5 = 1 and ID[5,0,4] = 'b000 ERDI[1:0] = 'b00, REI = 'b0, ERDI[1:0] and REI are encoded in the V5 byte.
K27.7+	-	b001001 0111	b00000111 Low order path frame alignment IV5 = 1 and ID[5,0,4] = 'b100 ERDI[1:0] = 'b00, REI = 'b1, ERDI[1:0] and REI are encoded in the V5 byte.
K28.7-	b001111 1000	-	b00001000 Low order path frame alignment ERDI[1:0] = 'b01, REI = 'b0, ERDI[1:0] and REI are encoded in the V5 byte.
K28.7+	-	b110000 0111	b00001001 Low order path frame alignment IV5 = 1 and ID[5,0,4] = 'b101 ERDI[1:0] = 'b01, REI = 'b0, ERDI[1:0] and REI are encoded in the V5 byte.
K29.7-	b101110 1000	-	b00001010 Low order path frame alignment

			IV5 = 1 and ID[5,0,4] = 'b010 ERDI[1:0] = 'b10, REI = 'b0, ERDI[1:0] and REI are encoded in the V5 byte.
K29.7+	-	b010001 0111	b00001011 Low order path frame alignment IV5 = 1 and ID[5,0,4] = 'b010 ERDI[1:0] = 'b10, REI = 'b1, ERDI[1:0] and REI are encoded in the V5 byte.
K30.7-	b011110 1000	-	b00001100 Low order path frame alignment IV5 = 1 and ID[5,0,4] = 'b011 ERDI[1:0] = 'b11, REI = 'b0, ERDI[1:0] and REI are encoded in the V5 byte.
K30.7+	-	b100001 0111	b00001101 Low order path frame alignment IV5 = 1 and ID[5,0,4] = 'b111 ERDI[1:0] = 'b11, REI = 'b1, ERDI[1:0] and REI are encoded in the V5 byte.
K23.7	b111010 1000	b000101 0111	b00001110 Non low-order path payload bytes ITPL = 0
Characters that Generate a Line Code Violation			
LCV	-	-	b00001111

The 8B/10B decoder generates a line code violation (LCV) signal. The RSEF maintains a 8-bit count of the total number of LCV's that have been received. The 8 bit counter value is accessible by polling ECBI register 02h (see section **Error! Reference source not found.**). A LCV occurs when the incoming data is not one of the valid 8B/10B characters or when the running disparity is incorrect. The TeleCombus special characters K28.0-, K28.0+, K28.4-, K27.7-, K27.7+, K28.7-, K28.7+, K29.7-, K29.7+, K30.7-, K30.7+, and K28.4+ are not considered LCV's but do effect the current running disparity calculation.

For received characters that generate a line code violation when decoded, the 8B/10B decoder outputs h0F and asserts K high.

The 8B/10B decoder is enabled and disabled by the EBTBEN mode pin. When disabled, the 8B/10B decoder passes input data to the output without modification and K is asserted low.

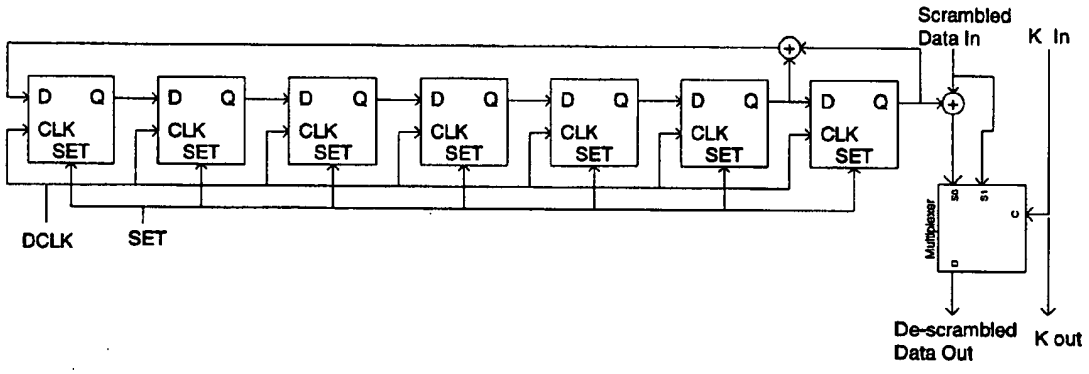
1.2.5.5 De-scrambler Block

The function of the de-scrambler block is to de-scramble the output of the 8B/10B decoder block. The de-scrambler block uses a frame synchronous de-scrambler that resets to all ones on the most significant bit of the first byte following the last J0/Z0 byte. The de-scrambler runs from that bit throughout the remainder of the frame. The A1, A2 and J0/Z0 bytes are not de-scrambled. The generating polynomial is $1 + x^6 + x^7$ and the sequence length is 127. Figure 17 represents the equivalent frame synchronous de-scrambler for a serial bit stream for illustrative purposes. The implementation de-scrambles data in parallel one byte at a time.

The descrambler is enabled and disabled by the DESCREN mode pin. When disabled, the descrambler passes data and K arriving at its input without modification.

When the descrambler is enabled, data is descrambled depending on the value of K. Data is descrambled when K is low and is passed without modification when K is high. The scrambler advances regardless of whether a byte is descrambled or not.

Figure 17 - Equivalent frame synchronous de-scrambler

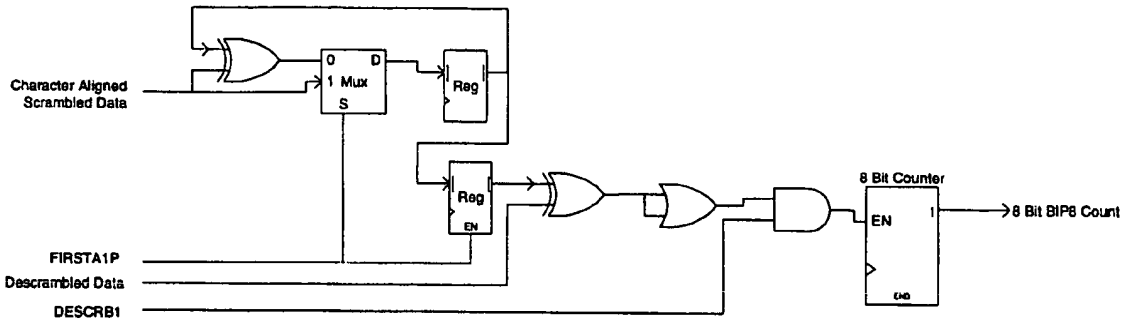


1.2.5.6 BIP8 Block

The BIP8 block calculates the BIP8 parity over a scrambled frame of data and modulo-2 adds the result to the descrambled B1 byte of the following frame. If the result is not zero (i.e. even parity) then the 8 bit BIP8 count is incremented and an interrupt is generated if enabled. The 8 bit count is accessible from the ECBI

LCVBIP8 count register. The BIP8 calculation and comparison are performed as shown in Figure 18.

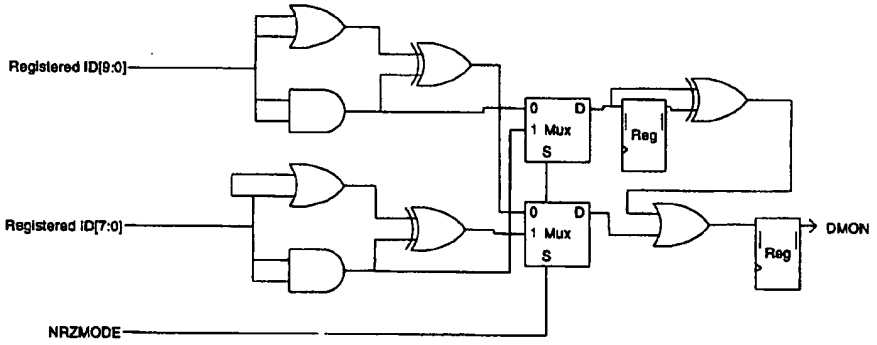
Figure 18 - BIP8 Check



1.2.5.7 Activity Monitor Block

The clock and data inputs on the RSEF are monitored for activity. The CLK_MON bit of the ECBI Interrupt Status and Activity Monitor register will be set high on the first rising edge of the input clock after reset. Refer to Figure 19 for the circuit used to generate the DMON status bit of the ECBI Interrupt Status and Activity Monitor register.

Figure 19 - Data Activity Monitor



The ID bus is monitored after being registered. Ten bits are monitored in 8b/10b framing mode while 8 bits are monitored in SNRZ framing mode. The DMON status bit will be asserted high if the data activity monitor detects bit transitions within 20 consecutive bits in 8b/10b framing mode or 16 consecutive bits in SNRZ framing mode.

1.2.6 OPERATION

1.2.6.1 Start-up Considerations

1.2.6.1.1 Interrupts

The RSEF has four interrupts, line code violation (LCVI), out-of-frame alignment (OFAI), BIP8 violation (BIP8I) and out-of-character alignment (OCAI). On startup, all the interrupts are masked by default.

Since the initial RSEF input data is not character aligned, it is recommended that the LCVI and BIP8I interrupts remain masked until character alignment has been achieved. This will avoid repeated LCVI interrupts.

1.2.6.1.2 Initial Character and Frame Alignment

Data leaving the RSEF should not be considered valid until the frame after the first frame that the RSEF found character and frame alignment.

1.2.6.1.3 Modes

The six RSEF mode pins (NRZMODE, STSNMODE[1:0], DIAGSHORT, EBTBEN, DESCREN and WCIMODE) are not expected to change during operation of the RSEF. These pins should be held in the desired state before the RSEF is reset.

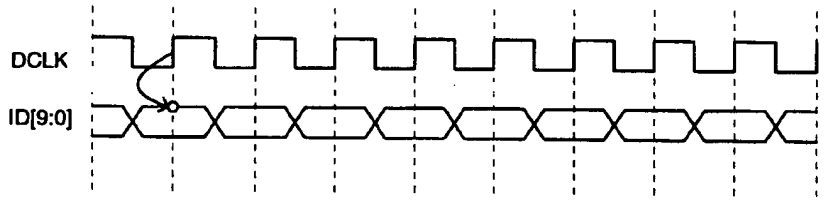
1.2.7 FUNCTIONAL TIMING

This section shows the functional relationship between inputs and outputs. No propagation delays are shown.

1.2.7.1 Input Timing

The unaligned input data is sampled on the rising edge of DCLK.

Figure 20 - Input Timing



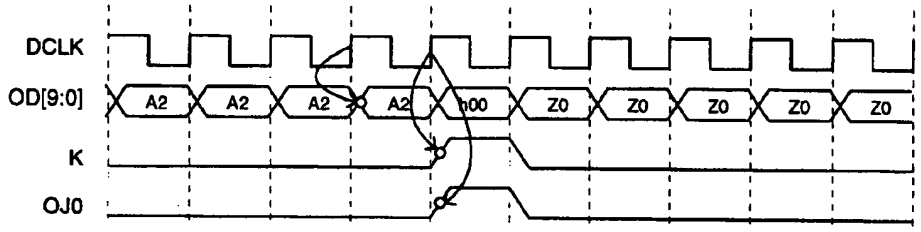
1.2.7.2 Output Timing

1.2.7.2.1 OD, OJ0 and K Output Timing

The K signal marks the appearance of control characters in the decoded 8B/10B input data. The OJ0 output marks the location of the J0 byte in the output data for both 8B/10B and scrambled NRZ framing modes.

The output data is asserted on the rising edge of DCLK. The timing output timing diagram of Figure 21 shows the output of the A2 framing characters, followed by the control character that marks the transport frame alignment signal, followed by the Z0 bytes when the 8B/10B decoder is enabled and when in 8B/10B framing mode. All data is output synchronously with respect to the input clock (DCLK) as shown in the figure. Note that the J0 character is substituted with h00 and K asserted high as per Table 13.

Figure 21 - Output Timing



1.2.8 IMPLEMENTATION DESCRIPTION

1.2.8.1 Design Capture

The RSEF was synthesized from VHDL-87 RTL. The design uses the standard ECBI block created by the CAD department. With the exception of the ECBI, all blocks use DCLK and are synchronous.

The following sections will outline the functionality of the design on a block-by-block basis. Each block in the design is addressed in a separate section. The name of each section corresponds to the name of a VHDL entity. When diagrams are included, the block labels correspond to the VHDL entity names. The block diagrams are intended to help explain functionality of the blocks. The name of signals and ports is not intended to correspond to the VHDL signal and port names. Mode and ECBI signals are generally left out of diagrams.

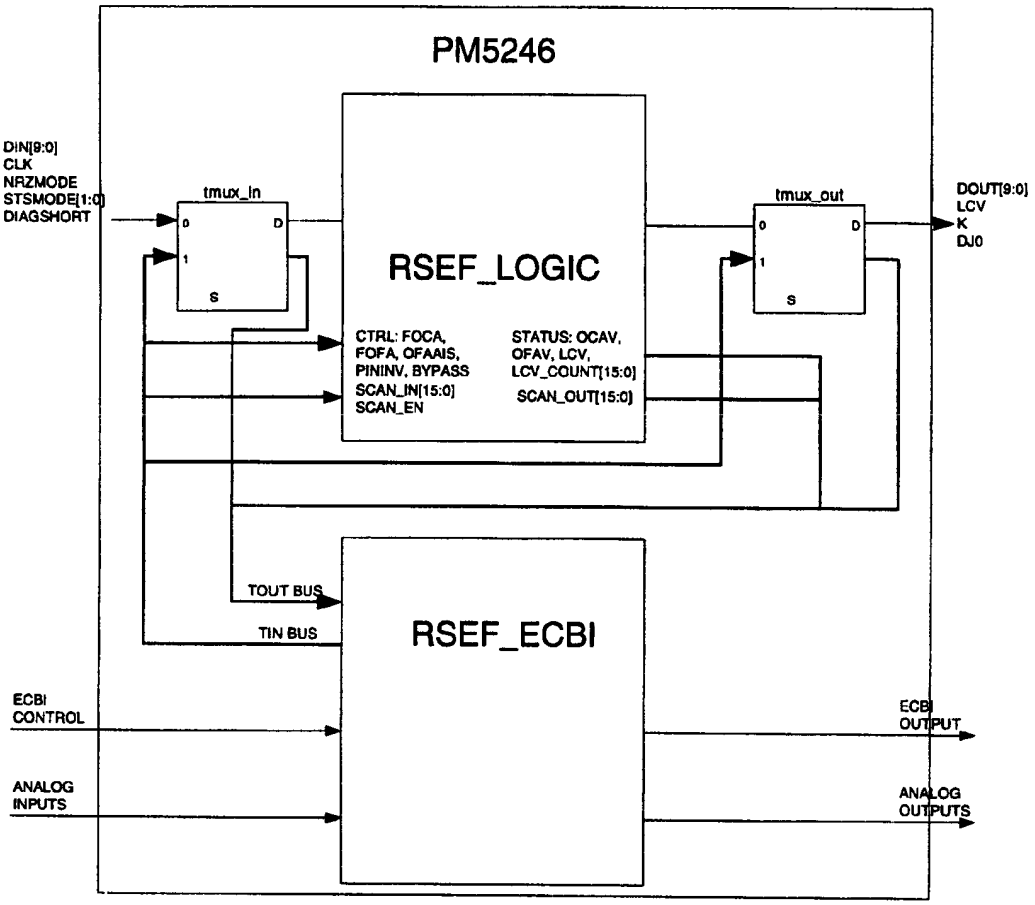
1.2.8.2 Top Level (PM5246)

Figure 22 gives a high level view of the first level of the hierarchy. The top level consists of two blocks: the RSEF_LOGIC block and the RSEF_ECBI block. Additional circuitry is used to allow control and observation of the inputs and outputs during scan.

The RSEF_ECBI block is a RSEF specific instantiation of the standard ECBI TSB provided by CAD. When in scan mode, the RSEF_ECBI controls the inputs and reads the outputs of the RSEF_LOGIC block, and controls and reads the scan_in and scan_out busses.

The RSEF_LOGIC block performs the character alignment, framing, and decoding of the unaligned 8B/10B or SNRZ input data. The block generates the aligned data, K, LCV, and OJ0 signals as described in the functional description.

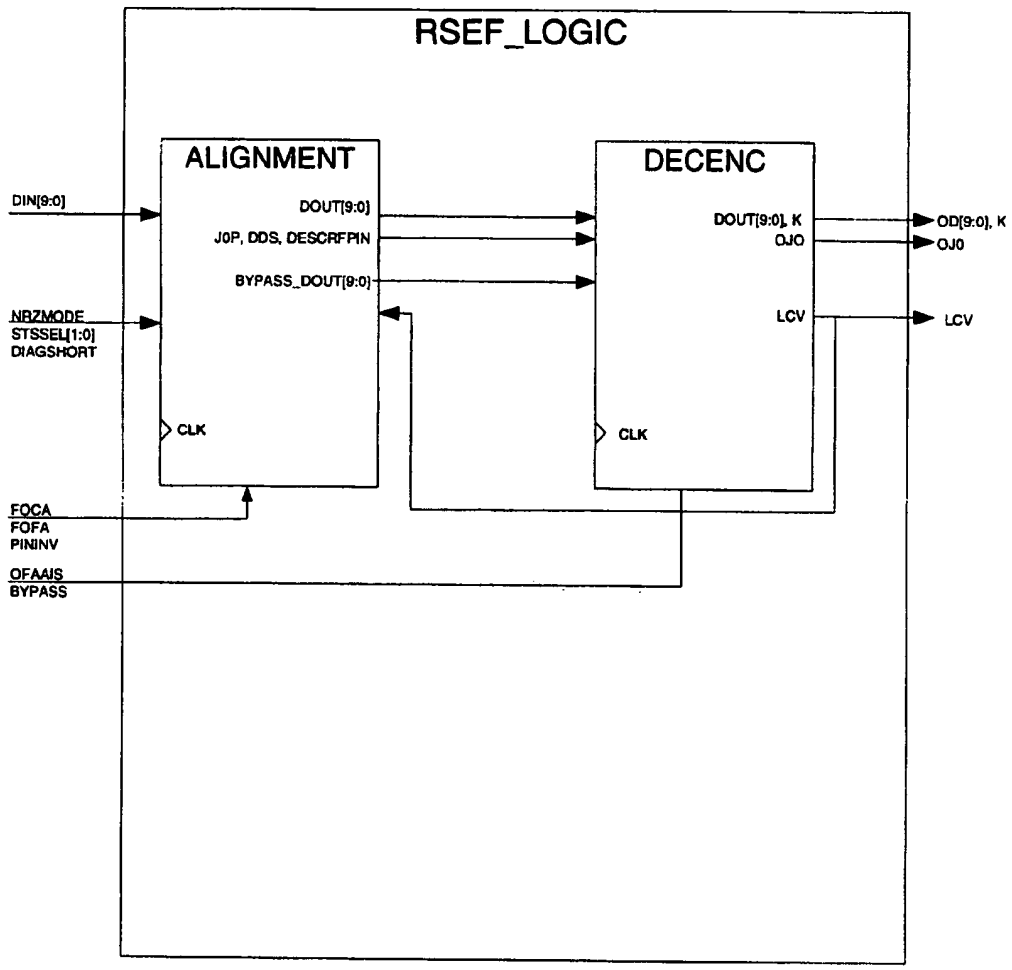
Figure 22 - Top Level Block Diagram



1.2.8.3 RSEF_LOGIC Block

The RSEF_LOGIC block diagram (Figure 23) contains two blocks. The ALIGN block performs character and frame alignment. The DECENC block performs 8B/10B and / or SNRZ decoding and then encodes the decoded result into the internal data format defined by Table 13.

Figure 23 - RSEF_LOGIC Block Diagram



1.3 eLVDS

The first implementation of the serial protocol described in this patent disclosure uses eLVDS as a physical layer realization. eLVDS is a .PMC-Sierra, Inc. standard for high-speed differential signaling. eLVDS is described in a separate patent disclosure. ELVDS is only one of many possible choices for physical layer transmission for this invention; the ideas described herein apply regardless of the choice of physical layer implementation.

1.4 Frame Alignment in a Multi-device Environment

The RJ0FP frame pulse is used to synchronise a set of devices that are inter-connected via LVDS links. It is provided concurrently to all the devices in the system once every 125µs, or multiples thereof. Characters retrieved from the receive LVDS links are written into a FIFO. When the J0 character is received, it is written into a fixed location in the FIFO. Subsequent characters are written the locations following. At each device in the system, a software configurable counter is used to mark the point, relative to RJ0FP, where all its receive LVDS links are expected to have delivered their J0 character. As directed by the delay counter, the device will then read the fixed location where the J0 character is stored, thus synchronising all the receive LVDS links. Differential delays between and clock instabilities of LVDS links are absorbed by the FIFOs.

The following diagram illustrates the system level view of the flexible receiver alignment modes implemented by the logic described above and in Section 1.3.

Figure 24 - Flexible System Timing

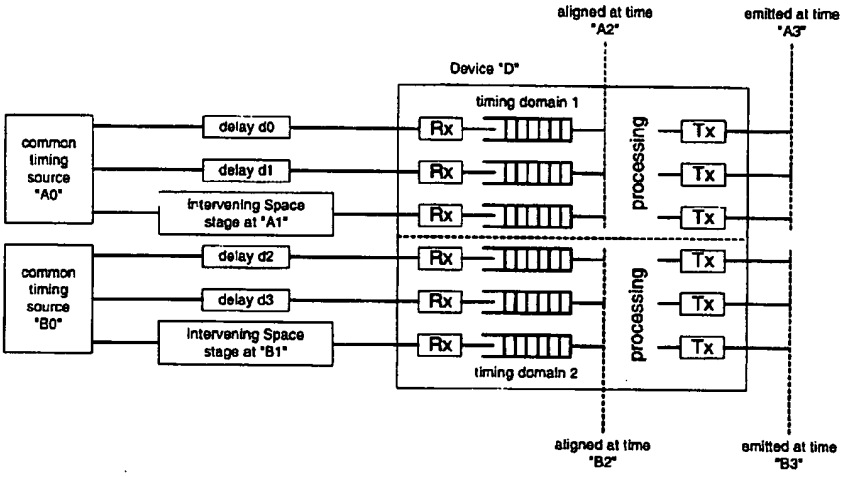


Figure 23 illustrates the flexible system-level timing achieved by the receiver logic, previously described. Figure 23 considers Device "D", which implements six receivers with their individual fifos. These receivers are divided into two separate timing domains (this division is optional, both timing domains could be collapsed into one common domain). All timing events occur relative to the global SONET Frame Pulse. The particular times (e.g., "A0") are delays from this global frame pulse.

Device D receives signals from two separate sources, one emitting its signals at time "A0", the other emitting its signals at time "B0".

The signals from "A0" follow three separate paths. The top two signals pass through differential delays on the routing of their PCB traces, and differential delays in the Tx and Rx logic. In addition, these signals may both be passed over optical links for part of their distance (by separate E-O and O-E devices), and the optical equipment and routing may add to their differential delays.

The Rx fifos described previously find the alignments for each of these signals, and ensure that both frames have arrived before any mutually aligned processing occurs.

The third signal from the source at time "A0" passes through a Space switching stage (with delay less than or equal to two SONET frame times, or 250 microseconds). This delay is much greater than the differential delay experienced by the top two signals from "A0".

The Rx fifos have sufficient depth to allow the signals on the top two paths to be delayed sufficiently to achieve mutual alignment with this third signal, and to allow full mutual alignment to occur within device D.

The lower three receivers on D receive signals from a separate source, at a separate time, "B0". These three signals have the same flexibility in alignment as described for the paths from "A0".

The additional feature illustrated by the presence of the signals from "B0" is that device D can be split into two separate timing domains, and can find separate alignment for the "A" and "B" signal paths. This feature allows device D to play two roles in a switching network, for instance, it can serve as both Stage 1 and Stage 3 in a "folded" three stage Clos network.

The following temporal equations provide additional information and constraints about these system timing features:

$$A0 < A1 < A2 < A3$$

$$B0 < B1 < B2 < B3$$

$$A2 - A0 < 250\mu s$$

$$B2 - B0 < 250\mu s$$

ALTERNATE EMBODIMENTS

The are many serial, differential signaling techniques which are closely related to eLVDS, standard LVDS, and CML; different choices at this level do not significantly change the present invention. Other standard and pseudo-standard

SONET rates could be supported (e.g. STS-3, or STS-52) without significantly changing the present invention. Other SONET/SDH overhead manipulations could be included without significantly altering the invetion. Other scrambling codes could be used without significantly altering the invention.